



Calibración de array cámaras ToF para reconstrucción volumétrica

**Máster Universitario en Sistemas Electrónicos Avanzados
Departamento de Electrónica**

**Presentado por:
D. Elena Rufo Merino**

**Dirigido por:
Dr. D. José Luis Lázaró Galilea
Dr. D. Alfredo Gardel Vicente**

Alcalá de Henares, a 22 de Mayo de 2013

AGRADECIMIENTOS

En primer lugar, querría agradecer a mis padres por la ayuda que me han brindado en todo momento, apoyándome y haciendo todo lo posible por facilitarme el camino, no solamente durante estos últimos meses, sino desde que tengo memoria. Gracias por animarme y motivarme siempre, sin vosotros nunca habría llegado hasta aquí.

También querría agradecer a José Luis Lázaro por confiar en mí, por brindarme la oportunidad de dedicarme a lo que realmente me gusta e introducirme en el mundo de la investigación. Por compartir conmigo su experiencia y sus conocimientos y haber hecho posible este Trabajo.

A mis compañeros de laboratorio y demás doctorandos de GEINTRA, ya que gracias a su compañía los días pasan mucho más rápido y siempre son entretenidos, y hasta una tarde de viernes en la universidad se puede convertir en el deseado día del helado. Y más en especial, gracias a David R, David S y Jorge por su ayuda tanto en los aspectos técnicos, como en lo personal, y por hacerme siempre sacar una sonrisa.

Cómo no, agradecer a los que han sido mis amigos durante todos mis años de Universidad, por cuidarme y estar a mi lado siempre, también por escuchar mi chiste de los espaguetis, por ponerme estados bonitos en Facebook, y por tantos momentos que hemos compartido. A Fran, por ser el bonobo alfa vetado, por esos maravillosos tiempos en la biblioteca compartiendo desesperaciones y alegrías. A Javi, porque nos encanta llevarnos la contraria, pero en el fondo siempre llevo razón yo, gracias por tu apoyo incondicional. A Rodri, por todas esas horas de charlas y quejándonos del mundo, por entenderme y aconsejarme siempre.

Y por último, pero no menos importante, al resto de mi familia y amigos que forman parte de mi vida, que han puesto su granito de arena y han caminado conmigo.

Resumen

El Trabajo de Fin de Máster que se desarrolla a continuación se enmarca dentro del ámbito de la visión computacional y propone un sistema robusto de calibración de un sistema formado por cuatro cámaras de tiempo de vuelo (ToF – Time of Flight) de bajo coste, pero a la vez de muy baja resolución (120 x 160 píxeles). El objetivo es conseguir un sistema económico para reconstrucción volumétrica de una escena fija. Para la consecución de este objetivo es necesaria la búsqueda de un patrón cuyos puntos característicos sean fácilmente detectables en las imágenes, que además no son de escala de grises, sino que son obtenidas con el mismo sensor ToF, así como el diseño de un algoritmo de procesamiento de imagen que logre dicho objetivo con precisión por debajo del píxel para lograr resultados de calibración óptimos. Además, también se realiza una calibración estéreo para conocer la posición relativa de las cámaras entre sí y poder relacionar los datos de cada una de ellas. Como último objetivo, se ha realizado una reconstrucción tridimensional en la que se fusionen los datos de todas las cámaras pertenecientes al sistema, estudiando los errores que se producen, a qué se deben y cómo pueden corregirse.

Para realizar estos objetivos se va a utilizar la herramienta software MATLAB®, partiendo de la toolbox de calibración de Yves-Bouguet (ampliamente utilizada en calibración de cámaras), estudiando su validez para cámaras de baja resolución de este tipo y desarrollando los algoritmos complementarios necesarios para poder concluir en unos resultados de calibración óptimos y la visualización interactiva de las superficies tridimensionales obtenidas. Pero dadas las características de los sensores utilizados, no todas las herramientas disponibles (en MATLAB u otros paquetes software) han permitido calibrar dichos sensores, por lo que existen partes del proceso de calibración para las que se han debido desarrollar herramientas propias, y en otros casos ha habido que estudiar cómo aportar los datos para usar únicamente parte de ellas.

Abstract

The developed Master Thesis is related to the computational vision field and it aims to robustly calibrate a four low cost Time of Flight (ToF) camera system, marked by their low quality and very low resolution (120 x 160 pixels), trying to reach a final volumetric reconstruction of the viewed scene. In order to achieve this goal, it is required to search a desirable pattern (chessboard, dotted...) whose feature points are easily detected in the acquired images (which are not gray scale images, they are obtained from the ToF sensor itself). Moreover, it is essential to develop an algorithm able to detect control points with sub-pixel accuracy in low resolution images if optimal calibration data must be calculated. Moreover, a stereo calibration process it is also performed so as to get to know the fixed relative positions and orientations between the 4 cameras set and being able to relate data obtained from each of them. As an ultimate purpose, data from all the cameras belonging to the full system is related to the same coordinate system through stereo calibration results and also merged in one single mesh in order to perform and visualize a 3D reconstructed figure of the scene, analyzing the existing errors, why they are originated and how they could be corrected.

In order to accomplish this targets, MATLAB® software is going to be used, and mainly the Yves-Bouguet calibration toolbox (widely accepted in general camera calibration applications), performing a study about its validity with this kind of low resolution cameras and also developing complementary algorithms required to finally conclude in optimal calibration results and visualize 3D meshes obtained

Índice

1. Introducción	21
1.1. Objetivos	22
1.2. Estructura de la memoria	23
2. Descripción del sistema	25
2.1. Sistema Propuesto	25
2.2. Cámaras de tiempo de vuelo	27
2.3. Modelo Pinhole y parámetros de distorsión	30
2.4. Necesidad de calibrar una cámara ToF	34
2.5. Descripción del sensor ToF y algoritmo de calibración	35
2.5.1. Selección de la cámara ToF	35
2.5.2. Calibración	37
3. Extracción de puntos característicos para la calibración de la cámara ToF	39
3.1. Precisión sub-pixélica	41
3.2. Elección del patrón	42
3.3. Búsqueda de las elipses y cálculo de los centroides	45
3.3.1. Filtro ‘unsharp’	47
3.3.2. Detector de bordes ‘canny’	49
3.3.3. Selección de formas elípticas	50
3.3.4. Búsqueda de radio adaptativo	52
3.3.5. Cálculo sub-pixélico del centro de las elipses	53
3.4. Ordenación de centroides	56
4. Calibración de las cámaras	61
4.1. Método Zhang	61
4.2. Toolbox Yves-Bouguet MATLAB®	62
4.3. Comparación con distintos patrones	63
4.3.1. Medida de la linealidad	63
4.3.2. Comparación de resultados con distintos patrones mediante linealidad	67
5. Calibración estéreo de cámaras ToF	73
5.1. Sistema estéreo. Geometría epipolar	74
5.1.1. Vector de traslación y matriz de rotación. Ángulos de Euler	75
5.2. Sistema de calibración	76
5.2.1. Obtención de imágenes estéreo y puntos característicos	77
5.2.2. Algoritmo ‘Rodrigues’	77

5.2.3. Calibración mediante MATLAB®	79
5.3. Reconstrucción mediante triangulación	82
5.4. Comprobación de la calibración de los 4 pares estéreo de cámaras ToF	83
5.5. Relación de los 4 pares de calibración estéreo	86
6. Reconstrucción 3D	89
6.1. Reconstrucción 3D de una sola cámara	89
6.1.1. Eliminación de paredes	90
6.1.2. Análisis de precisión	92
6.2. Fusión de datos de dos cámaras	94
6.3. Reconstrucción 3D con las cuatro cámaras	101
7. Toolbox realizada	103
7.1. Organización del directorio de la interfaz	103
7.2. Calibración de las cámaras	104
7.2.1. Nueva calibración de una cámara	105
7.2.2. Nueva calibración estéreo	106
7.3. Obtener datos en 3D de una de las cámaras	107
7.4. Representar las vistas de 2 de las cámaras	108
7.5. Visualizar la figura con las vistas de las 4 cámaras	110
8. Conclusiones y trabajos futuros	113
8.1. Conclusiones	113
8.2. Trabajos futuros	114
9. Bibliografía	115
ANEXO I- Detectores de bordes	117
ANEXO II: Método Zhang	119
II.1. Homografía entre el plano modelo y el plano imagen	119
II.2. Restricciones de los parámetros intrínsecos	119
II.3. Resolución de la calibración de la cámara	120
II.4. Estimación de máxima verosimilitud	122
ANEXO III. Geometría epipolar	123

Índice de Figuras

Figura 1. Entorno a reconstruir	21
Figura 2. Imagen del entorno elaborado para el desarrollo del sistema de reconstrucción	25
Figura 3. Sistema propuesto	26
Figura 4. Distintos métodos de medida de distancia sin contacto.....	27
Figura 5. Izquierda: imagen en escala de grises. Derecha: imagen de distancias de una cámara ToF.....	28
Figura 6. Principio de medidas ToF. Medida de tiempo.	28
Figura 7. Principio de medidas ToF. Medida de desfase.	29
Figura 8. Señal emitida (azul) y señal recibida (rojo)	29
Figura 9. Modelo pin-hole de una cámara.....	30
Figura 10. Correspondencia de parámetros de la matriz intrínseca con la estructura del sensor CMOS	31
Figura 11. Rotación y traslación aplicada	32
Figura 12. Efecto de la distorsión radial	33
Figura 13. Efecto de la distorsión radial. Imagen sin distorsión (izquierda), con distorsión de barril (centro) y de cojín (derecha).....	33
Figura 14. Coordenadas tridimensionales y distancia radial	34
Figura 15. Patrón de la cámara Kinect proyectado sobre una escena	36
Figura 16. CamBoard Nano	36
Figura 17. Patrón utilizado para la calibración.....	39
Figura 18. Imagen de un patrón en el que se producen reflexiones	39
Figura 19. Estimación sub-pixélica de una curva.....	41
Figura 20. Resultado de aplicar la PSF al objeto inicial ideal.	41
Figura 21. El valor del píxel es hallado mediante interpolación con los píxeles próximos.	42
Figura 22. Patrón de tablero de ajedrez para detectar esquinas.....	43
Figura 23. Búsqueda de esquinas con MATLAB®.....	43
Figura 24. Posiciones del damero en las que no se ha podido encontrar las esquinas.	43
Figura 25. Patrones damero de mayor tamaño de cuadros.....	44
Figura 26. Patrones de calibración circulares. Patrón elegido (derecha)	45
Figura 27. Distintas imágenes tomadas con el patrón en distintas posiciones	45
Figura 28. Flujograma de la búsqueda de elipses y cálculo de posición de los centroides	46
Figura 29. Filtro unsharp. Imagen inicial (arriba) e imagen tras la aplicación del filtro (abajo) ..	47
Figura 30. Bordes en la imagen inicial (izquierda), tras aplicar el filtro <i>unsharp</i> (centro) y transición en los bordes (real en verde e ideal en negro)	48
Figura 31. Imagen del patrón inicial (izquierda) y tras aplicar el filtro unsharp (derecha).....	48
Figura 32. Bordes según los métodos explicados en una imagen cercana.	49
Figura 33. Contornos abiertos en la detección de bordes.....	49
Figura 34. Bordes que juntan los contornos	50
Figura 35. Distintos valores de excentricidad de elipses.....	50
Figura 36. Objeto cerrado $E = 0$ (izquierda) y abierto $E = 1$ (derecha).....	51

Figura 37. Imagen de bordes (izquierda) e imagen que contiene solo las elipses del patrón (derecha).....	51
Figura 38. Imagen de bordes inicial (izquierda), elipses detectadas inicialmente (centro) y elipses detectadas tras el algoritmo de eliminación de elipses sobrantes (derecha).....	52
Figura 39. Imagen sobre la que hallar centros (izquierda), centros hallados con un radio fijo incorrecto (centro) y centros hallados con un radio adaptativo (derecha).....	52
Figura 40. Imagen capturada en la que el patrón presenta una gran inclinación con respecto a la cámara	53
Figura 41. Ejemplo de patrón con cálculo de centroides con radio adaptativo.	53
Figura 42. Elipse inicial (izquierda) y elipse tras interpolación (derecha).....	54
Figura 43. Gradientes en X e Y	54
Figura 44. Cálculo de centroides. Centroides iniciales (izquierda) y tras el cálculo sub-pixélico	55
Figura 45. Vecindad tomada para cálculo de centroide (izquierda) y cálculo sub-pixélico (derecha).....	55
Figura 46. Orden en el que se deben almacenar los centroides encontrados	57
Figura 47. Ordenación.....	57
Figura 48. Paso 1	57
Figura 49. Paso 2.....	57
Figura 50. Paso 3.....	57
Figura 51. Paso 4.....	58
Figura 52. Paso 5.....	58
Figura 53. Paso 6.....	58
Figura 54. Paso 7.....	58
Figura 55. Paso 8.....	58
Figura 56. Paso final	58
Figura 57. Ejemplo de colocación del patrón en el que el centroide 5 está más cerca que el 4..	59
Figura 58. Centroides ordenados en varias imágenes	59
Figura 59. Interfaz gráfica de la toolbox Yves-Bouguet.....	62
Figura 60. Error de reproyección en píxeles (izquierda) y patrones en 3D (derecha).....	63
Figura 61. Ejemplo de pares de imágenes reales con distorsión (arriba) y su correspondiente imagen con la distorsión corregida (abajo)	65
Figura 62. Centroides con y sin distorsión representados sobre una imagen capturada	65
Figura 63. Cálculo de linealidad. Rectas horizontales (izquierda) y verticales (derecha).....	66
Figura 64. Resultados de los parámetros de algunos ejemplos de calibración con distintos métodos y patrones.....	68
Figura 65. Patrones utilizados para la calibración. De izquierda a derecha: damero grande, damero pequeño, círculos pequeños y círculos grandes.....	68
Figura 66. Distintas correcciones de distorsión los parámetros hallados por cada método (1-6) de izquierda a derecha y de arriba a abajo	68
Figura 67. Tabla de datos de coeficientes de linealidad.....	70
Figura 68. Proyección de dos puntos 3D (P y Q) en 2D (P' y Q').....	73
Figura 69. Sistema estéreo con cámara izquierda y derecha	74
Figura 70. Ángulos de Euler	75
Figura 71. Imágenes estéreo de la cámara 1 y 2 con los centroides marcados.....	77
Figura 72. Operaciones realizadas por el método Rodrigues	78

Figura 73. Toolbox de calibración estéreo	79
Figura 74. Proceso inicial de calibración	80
Figura 75. Posición relativa entre cámaras 1 y 2 y los centroides usados para calibrar	81
Figura 76. Sistemas de referencia de cada par de cámaras tras la calibración estéreo. De izquierda a derecha, pares de cámaras 1-2, 2-3, 2-4 y 4-5	81
Figura 77. Imágenes estéreo del patrón en distintas posiciones de la escena con las cámaras 1 y 2	83
Figura 78. Reconstrucción 3D de los centroides de los patrones de cada una de las imágenes de la Figura 77	83
Figura 79. Distancias (en centímetros) en los centroides re-proyectados mediante los datos de calibración estéreo de las cámaras 1-2	84
Figura 80. Imágenes estéreo del patrón en distintas posiciones de la escena con las cámaras 2 y 3	85
Figura 81. Reconstrucción 3D de los centroides de los patrones de cada una de las imágenes de la Figura 80	85
Figura 82. Distancias (en centímetros) en los centroides re-proyectados mediante los datos de calibración estéreo de las cámaras 2-3	85
Figura 83. Distancias (en centímetros) en los centroides re-proyectados mediante los datos de calibración estéreo de las cámaras 3-4	86
Figura 84. Distancias (en centímetros) en los centroides re-proyectados mediante los datos de calibración estéreo de las cámaras 4-1	86
Figura 85. Sistema completo a partir de las calibraciones estéreo	87
Figura 86. Distancias reales y calculadas entre las cámaras del sistema.....	88
Figura 87. Reconstrucción 3D individual de un cubo por cada cámara. De izquierda a derecha, cámaras 1, 2, 3 y 4. De arriba abajo, imagen de amplitudes, reconstrucción 3D con escala de distancias y reconstrucción 3D con amplitudes.	89
Figura 88. Reconstrucción 3D individual de una escena más compleja por cada cámara. De izquierda a derecha, cámaras 1, 2, 3 y 4. De arriba abajo, imagen de amplitudes, reconstrucción 3D con escala de distancias y reconstrucción 3D con amplitudes.	90
Figura 89. Superficie 3D del cubo con las paredes eliminadas.....	91
Figura 90. Superficie 3D de la escena más compleja con las paredes eliminadas	91
Figura 91. Superficie 3D del cubo con las paredes eliminadas y la pseudo escala de grises añadida	91
Figura 92. Superficie 3D de la escena más compleja con las paredes eliminadas y la pseudo escala de grises añadida.....	91
Figura 93. Imagen de pseudo-escala de grises (izquierda) y de distancias (derecha).	92
Figura 94. Puntos de los cuales se está midiendo la distancia.....	92
Figura 95. Comparación de distancias	93
Figura 96. Reconstrucción 3D de una caja rectangular situada en la escena.	93
Figura 97. Comprobación de distancias tras la corrección del factor multiplicativo	94
Figura 98. Imagen de la escena a reconstruir desde la cámara 1(izquierda) y desde la cámara 2 (derecha).....	94
Figura 99. Superficie 3D de la cámara 1 (izquierda) y de la cámara 2 (derecha). Problema superficies ocultas.	95
Figura 100. Representación 3D de la nube de puntos de la cámara 1 (izquierda) y la cámara 2 (derecha).....	95

Figura 101. Superficies de la cámara 1 y 2 referidas al mismo sistema de coordenadas mediante aplicación de los parámetros del sistema estéreo.	96
Figura 102. Nubes de puntos de la cámara 1 y 2 referidas al mismo sistema de coordenadas mediante aplicación de los parámetros del sistema estéreo.	96
Figura 103. Fusión de datos de las dos nubes estéreo.	96
Figura 104. Fusión de datos de las dos nubes estéreo con escala de distancias	97
Figura 105. Tabla de errores de distancia en la reconstrucción 3D estéreo.	97
Figura 106. Imagen de la escena a reconstruir desde la cámara 3 (izquierda) y desde la cámara 4 (derecha).....	98
Figura 107. Superficie 3D de la cámara 3 (izquierda) y de la cámara 4 (derecha).	98
Figura 108. Superficies de la cámara 3 y 4 referidas al mismo sistema de coordenadas mediante aplicación de los parámetros del sistema estéreo.	98
Figura 109. Fusión de datos de las dos nubes estéreo	99
Figura 110. Nubes de puntos alineadas tras ejecutar Iterative Closest Points.....	99
Figura 111. Reconstrucción 3D final de la pirámide con las cámaras 3 y 4	99
Figura 112. Tabla de comparación entre distancias en la reconstrucción 3D y medidas reales	100
Figura 113. De izquierda a derecha, imágenes de amplitud de la escena a reconstruir desde la cámara 1 a la 4.....	101
Figura 114. De izquierda a derecha y de arriba abajo, superficies 3D de las cámaras 1,2,3 y 4.	101
Figura 115. Superficies de las cuatro cámaras referidas al mismo sistema de coordenadas mediante aplicación de los parámetros de los cuatro pares estéreo.....	101
Figura 116. Figura 115 con las zonas de las paredes minimizadas	102
Figura 117. Arriba: cubo reconstruido. Abajo: cubo reconstruido con respecto a las cámaras desde dos perspectivas distintas	102
Figura 118. Directorio con carpetas y subcarpetas necesarias para el uso de la toolbox	103
Figura 119. Menú principal de la interfaz.	104
Figura 120. Menú de calibración.....	104
Figura 121. Menú de carga de calibración	104
Figura 122. Selección del directorio para cargar los datos de calibración	105
Figura 123. Selección de las imágenes de calibración	105
Figura 124. Extracción de puntos característicos en una imagen.....	106
Figura 125. Click en centroide inicial (izquierda) y centroides ordenados (derecha).....	106
Figura 126. Resultados de una calibración estéreo	107
Figura 127. Menú de obtención de datos de las cámaras	107
Figura 128. Imagen de distancias (izquierda) y pseudo-imagen de escala de grises (derecha). 107	
Figura 129. Superficie tridimensional obtenida (izquierda) y con nivel de gris añadido (derecha)	108
Figura 130. Modificaciones sobre los puntos 3D obtenidos por la cámara.....	108
Figura 131. Selección del par de cámaras del que se quiere tomar su nube de puntos	108
Figura 132. Proceso de selección de archivos necesario para reconstrucción 3D estéreo	109
Figura 133. Vista 3D de la escena obtenida por dos cámaras ToF. Izquierda: nube de puntos. Derecha: superficie continua.....	109
Figura 134. Vista 3D de dos cámaras tras aplicar ICP. Izquierda: nube de puntos. Derecha: superficie continua	109
Figura 135. Cargar en memoria los ficheros con los puntos 3D de cada cámara.....	110

Figura 136. Cargar en memoria los ficheros de calibración de cada cámara	110
Figura 137. Vista 3D de la escena de las 4 cámaras ToF. Izquierda: nube de puntos. Derecha: superficie continua	110
Figura 138. Vista 3D de las 4 cámaras tras aplicar ICP. Izquierda: nube de puntos. Derecha: superficie continua	111
Figura 139. Geometría epipolar	123

1. Introducción

La visión artificial es utilizada en multitud de ámbitos en la actualidad, como por ejemplo, en servicios de video-vigilancia, inspección visual, seguimiento o tracking de objetos en movimiento, procesamiento de imágenes médicas, robots controlados o guiados por visión, reconocimiento de objetos... En muchos de estos casos puede ser necesario, ya sea como primer paso del procesamiento de datos o como objetivo final, obtener la vista del entorno o elemento bajo estudio en tres dimensiones. Por lo tanto, un sistema de reconstrucción tridimensional pretende representar la información de los objetos situados en un entorno observado, permitiendo almacenar y tratar dichos datos para poder representarlos y realizar el procesamiento que se requiera mediante un computador. La reconstrucción en 3D es un gran reto hoy en día, ya que se pretende alcanzar un resultado preciso, fiable y robusto en el menor tiempo posible, tendiendo hacia sistemas en tiempo real.

El presente proyecto tiene como objetivo final reconstruir los objetos presentes en un escenario rectangular mediante cuatro cámaras de tiempo de vuelo, como se muestra en la Figura 1. La innovación que propone el sistema con respecto a la mayoría de los sistemas existentes es que, en lugar de realizar la reconstrucción con cámaras RGB o de escala de grises, obteniendo las coordenadas en 3D mediante triangulación gracias a las vistas estéreo, se pretende realizar mediante la unión de la información de distancia o profundidad que proporciona cada una de las cuatro cámaras de tiempo de vuelo.

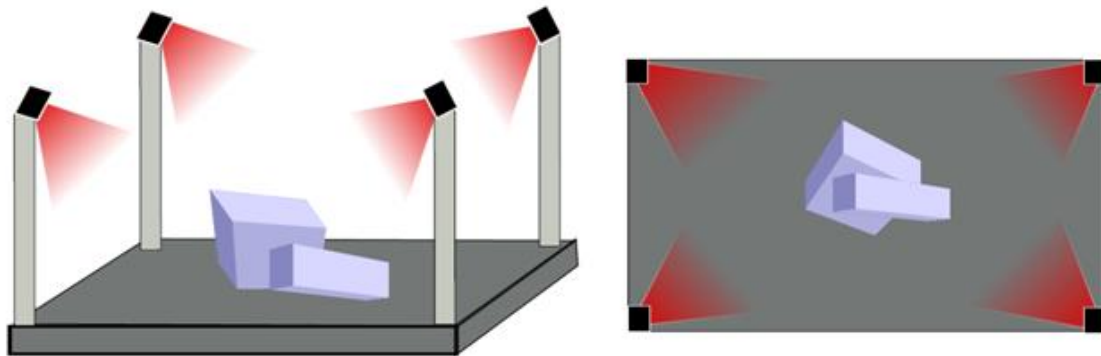


Figura 1. Entorno a reconstruir

Los sistemas de reconstrucción 3D mediante acoplamiento de varias vistas estéreo a menudo fallan si los objetos presentan zonas de oclusión, es decir, zonas que no son visibles desde la perspectiva de ambas cámaras. En consecuencia, solamente se pueden obtener las coordenadas 3D de aquellos puntos de las imágenes que estén en el campo de visión común de las dos cámaras (sin oclusión en ninguna de las dos vistas). Sin embargo, con una sola cámara de tiempo de vuelo (ToF – Time of Flight) se puede obtener información en 3D de toda la zona incluida en su campo de visión, por lo que el área que puede abarcarse con un sistema formado por cámaras ToF de reconstrucción 3D es mayor que con un número igual de cámaras RGB mediante visión estéreo y triangulación.

Una cámara ToF proporciona información sobre la distancia existente desde el punto de referencia del sensor hasta los puntos visibles de la escena. Sin embargo, para poder calcular las coordenadas en tres dimensiones de manera correcta y precisa, que es lo que realmente interesa

de cara a la reconstrucción de un objeto en 3D, es necesario realizar una calibración geométrica de la cámara que permita la conversión de distancia -a coordenadas (X, Y, Z) . Por lo tanto, se necesita realizar un proceso de calibración para obtener los parámetros intrínsecos y extrínsecos de cada una de las cámaras del sistema de manera individual, así como una calibración estéreo para obtener los parámetros de rotación y traslación de las cámaras dos a dos para conocer la posición relativa entre cada dos cámaras de las que forman el sistema y de esta manera poder superponer las distintas vistas obtenidas y realizar una reconstrucción tridimensional.

Además, este proyecto incorpora un reto añadido, que va a ser el que marca todas las características del proceso y del sistema: se va a realizar utilizando unas cámaras de tiempo de vuelo de muy bajo coste, las cuales tienen una resolución de imagen muy baja (120x160 píxeles) y poca precisión en la medida de distancias. Ya de por sí, debido a la resolución, se puede inducir un error de más de 1.5 cm en cada uno de los ejes X e Y por incertidumbres, y en cuanto a la medida de distancia, a pesar de que el fabricante solamente indica que está diseñada para precisión en la medida de distancias a corta distancia, en la realidad el error depende de la distancia real a la que se esté midiendo y del material de los objetos, llegando a alcanzar errores de más de 10 cm. Esta característica conlleva grandes problemas a la hora de detectar con precisión puntos característicos en las imágenes, lo cual es necesario como primera etapa para realizar el proceso de calibración. Debido a que se está utilizando únicamente un sensor de tiempo de vuelo, las imágenes utilizadas para la calibración no son imágenes en color o en escala de grises, que es con los que se suele contar para este proceso. En este caso, el propio sensor de profundidad, aparte de la propia información de distancia, devuelve también una imagen de amplitudes en blanco y negro. Sin embargo, es importante remarcar que no es una imagen en escala de grises como podría ser obtenida de cualquier otro sensor de imagen común, sino que es obtenida a través de la radiación infrarroja utilizada para hallar la distancia, tal y como se explica en el apartado 2.2, y las imágenes obtenidas no son como podríamos esperarse, ya que, por ejemplo, hay ciertos colores (como el rojo) o materiales que no se aprecian y se muestran en la imagen como si fueran un tono blanco, debido a que la amplitud con la que se represente un objeto dependerá de su reflectividad, en lo que no solo influye el color, sino el tipo de material del que esté fabricado.

El proceso se va a llevar a cabo con la herramienta MATLAB®. Debido a ello, se va a partir de la base de la Toolbox de calibración de cámaras más conocida y utilizada, la de Jean-Yves Bouguet¹, la cual se basa en el método Zhang [4] [5], y se va a observar que, a pesar de ser uno de los métodos más comúnmente usados, no siempre da un buen resultado, sobre todo en casos en los que se utilizan cámaras de baja resolución o que proporcionan imágenes de poca calidad y con alta distorsión.

1.1. Objetivos

El objetivo principal de este Trabajo Fin de Máster es el estudio de un método de calibración de un array de cámaras de tiempo de vuelo (cámaras ToF) de bajo coste y baja resolución que posibilite formar una escena a partir de la composición de las imágenes de todas las cámaras que formen el sistema. Para ello, es necesaria la búsqueda de un patrón apropiado que posibilite detectar en él con precisión sub-pixélica puntos característicos necesarios como base para la calibración.

¹ Camera Calibration Toolbox for Matlab - http://www.vision.caltech.edu/bouguetj/calib_doc/

A continuación, se ha de realizar una calibración de cada una de las cámaras del sistema de manera independiente y una calibración estéreo dos a dos con las cámaras contiguas del sistema. Para ello, se utilizan las funciones de calibración que proporciona la toolbox basada en el método Zhang de MATLAB®, *el cual no está específicamente diseñado para trabajar con imágenes de baja resolución y baja calidad*. Por lo tanto, otro de los objetivos del proyecto será estudiar la efectividad de este método tan comúnmente usado en cámaras de baja resolución.

Por último, se pretende tratar de reconstruir la escena en tres dimensiones a partir de los datos de calibración obtenidos en etapas anteriores, pudiendo visualizar los objetos situados en ella. Para mejorar la calidad de la nube de puntos compuesta de la vista de ambas cámaras habrá que utilizar un algoritmo iterativo que haga coincidir en mayor medida las cuatro nubes de puntos, para así obtener una reconstrucción lo más fiable posible.

Sin embargo, cabe destacar que debido a las cámaras de baja calidad y baja resolución que se van a utilizar para el desarrollo del proyecto van a aparecer una serie de inconvenientes, errores y problemas que no existen cuando se utilizan otros sistemas, como si se realizase la reconstrucción a través de cámaras RGB de mayor resolución, por ejemplo. Entre estos problemas cabe destacar que las imágenes de las que se parte no son RGB, ni de escala de grises, sino una pseudo-imagen de escala de grises obtenida a partir del método de medida de distancias lo que hace que algunos colores o materiales de la escena no se diferencien de otros, como podría ser esperado.

Por tanto, en primer lugar habrá problemas en la elección de un patrón válido para detectar puntos característicos a usar para la calibración, teniendo en cuenta que no funciona aceptablemente el reconocimiento de esquinas en un damero, tal y como está diseñado en OpenCV y MATLAB®. Por lo tanto, habrá que desechar algunas de las funciones de la toolbox de calibración, así como el estudio necesario de ellas para entender cómo devuelve los parámetros calculados y cómo hay que proporcionarle los datos a algunas de las funciones para que el resultado final sea el esperado. Con todo ello, se tendrá una variante de calibración para el sistema de cámaras abordado en el presente TFM.

1.2. Estructura de la memoria

La estructura del presente Trabajo Fin de Máster se ha dividido en las secciones que se enumeran a continuación:

- **Capítulo 2:** se realiza una breve descripción de las características del sistema propuesto para entender el contexto del trabajo.
- **Capítulo 3:** se aborda la elección de un patrón adecuado y la búsqueda y cálculo sub-pixélico de los puntos característicos necesarios para realizar la calibración de las cámaras.
- **Capítulo 4:** se analizan varios métodos posibles de calibración individual de las cámaras ToF, utilizando distintos patrones y funciones de búsqueda sub-pixélica de los puntos de control, comparándolos y eligiendo el que mejor resultados proporciona para continuar con el resto del proceso. Ha de elegirse un método que permita corregir la distorsión de las imágenes y obtener medidas tridimensionales lo más próximas a la realidad.

- **Capítulo 5:** en este capítulo se lleva a cabo la calibración estéreo de las cámaras dos a dos, con el objetivo de hallar las matrices de rotación y traslación que determinan la posición relativa entre ambas con cámaras de muy baja resolución.
- **Capítulo 6:** tras la calibración, se presenta el proceso de reconstrucción en tres dimensiones de la escena capturada, uniendo la información de las 4 cámaras del sistema, analizando también la efectividad de la calibración. También se explican posibles métodos de mejora de los resultados.
- **Capítulo 7:** se muestra la interfaz gráfica que ha sido diseñada para posibilitar de manera sencilla todo el proceso de obtención de puntos característicos, calibración, reconstrucción 3D y visualización de las imágenes.
- **Capítulo 8:** por último, se resumen las conclusiones obtenidas a lo largo de todo el trabajo y se presentan una serie de trabajos futuros por los que sería posible continuar con la investigación

2. Descripción del sistema

2.1. Sistema Propuesto

El objetivo final del proyecto es obtener una visualización en 3D de la escena bajo estudio. Por lo tanto, en primer lugar habrá que generar una escena controlada sobre la que se van a colocar las cámaras en una posición fija, ya que el más mínimo movimiento en las cámaras puede provocar que una calibración previamente realizada deje de ser válida. Por tanto, sobre una base rectangular de 57x86 cm, se colocan las cámaras elevadas 51 cm y orientadas hacia el centro de la escena, tal y como se observa en la Figura 2.



Figura 2. Imagen del entorno elaborado para el desarrollo del sistema de reconstrucción

Todas las pruebas y algoritmos desarrollados bajo este prototipo podrían posteriormente ser utilizados en aplicaciones diferentes con otras características, fondo variable, distancias más grandes, espacio no controlado, etc. Una vez establecido el entorno físico en el que se colocan las cámaras y los objetos a reconstruir, se deben definir las etapas necesarias que han de llevarse a cabo para conseguir los objetivos propuestos. Como se expresa esquemáticamente en la Figura 3, el proceso básicamente se puede dividir en una etapa de calibración de las cámaras y de acondicionamiento de datos para poder realizar la reconstrucción tridimensional.

En primer lugar, ha de realizarse una etapa de calibración de las cámaras de tiempo de vuelo a partir de las pseudo-imágenes de escala de grises que nos proporcionan. Este trabajo se centra principalmente en esta parte del proceso, la cual consta de dos fases principales:

- Calibración intrínseca: han de obtenerse los parámetros intrínsecos y de distorsión de cada una de las 4 cámaras utilizadas.
- Calibración estéreo: se hallan los parámetros de rotación y traslación de cada par de cámaras, de manera que se conozca la posición relativa existente entre ellas.

Cuando se haya realizado la calibración, han de obtenerse las coordenadas 3D de la escena de cada una de las cámaras, para lo cual serán necesarios los parámetros intrínsecos y de distorsión obtenidos en la primera etapa y las imágenes de profundidad de las cámaras ToF.

A continuación ha de fusionarse la información de las cuatro cámaras, y para ello hay que aplicar las matrices de rotación y traslación obtenidos en la calibración estéreo a las nubes

de puntos obtenidas por cada cámara. Sin embargo, los datos pueden no estar perfectamente alineados, y será necesario operar con dichas nubes de puntos para lograr que coincidan y conseguir un objeto 3D con una única superficie.

Por último, se añadirá la información de escala de grises que nos proporcionan las pseudo-imágenes en blanco y negro de las cámaras ToF para poder obtener una representación tridimensional más atractiva e intuitiva de la escena.

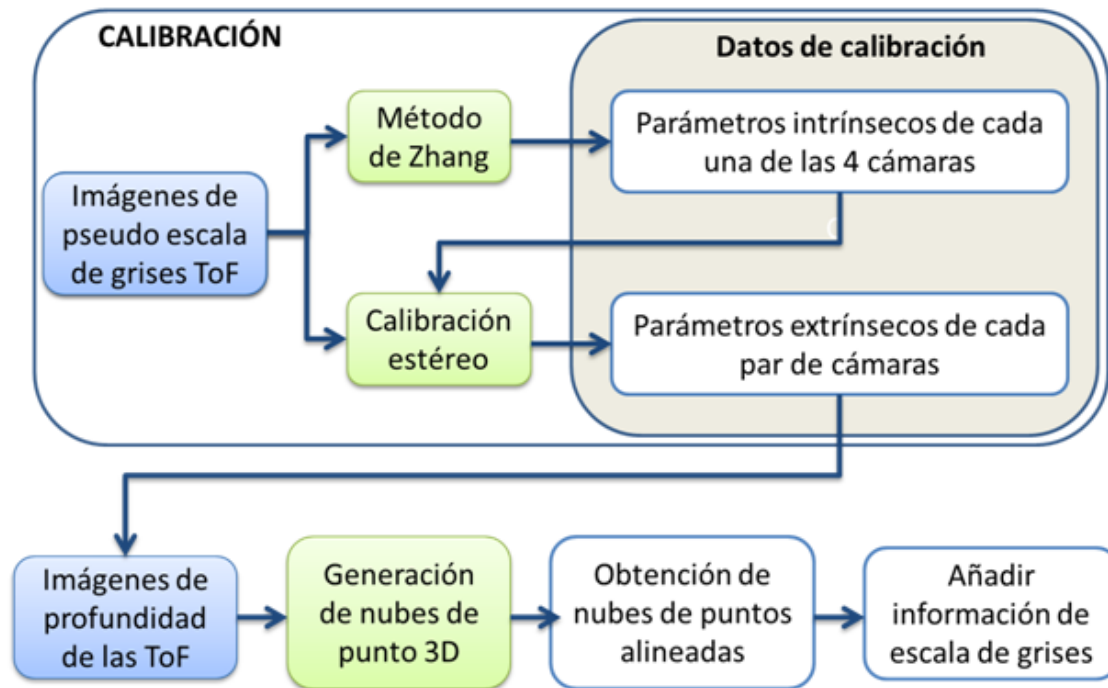


Figura 3. Sistema propuesto

Sin embargo, todo este proceso descrito no es tan trivial como puede parecer, ya que debido al tipo de cámaras que están siendo utilizadas se van a presentar errores e inconvenientes que han de tenerse en cuenta para lograr los objetivos finales. Por lo tanto, cabe destacar que en este proyecto no se trata de realizar una simple calibración de cámaras y una posterior reconstrucción 3D de la escena, sino que se pretenden estudiar y solventar los problemas existentes debido al uso de un sensor de tiempo de vuelo de muy baja resolución, y estudiar la validez del método de MATLAB® para este tipo de cámaras, entre ellos:

- Los algoritmos diseñados en la toolbox de MATLAB® para encontrar esquinas en patrones tipo damero no son válidos para alcanzar precisión en la búsqueda de los puntos de control en cámaras de tan baja resolución
- Las funciones análogas de búsqueda de esquinas programadas en OpenCV.
- La búsqueda de centroides, diseñando una función propia que permita la detección de estos puntos.
- Buscar una solución de compromiso entre el tamaño de las circunferencias del patrón para que se puedan diferenciar correctamente en las imágenes, y además que el número de puntos característicos sea suficiente para obtener resultados satisfactorios durante la calibración.

Debido a que solo algunas de las funciones de calibración MATLAB® se van a utilizar, va a ser necesario estudiar los parámetros intermedios que manejan, ya que en muchos de los casos no se corresponden con los que intuitivamente cabría esperar y es necesario realizar alguna modificación en los datos que hay que pasar o que devuelven algunas funciones de la toolbox. Si se utiliza la toolbox de principio a fin y no se deben realizar intervenciones entre distintas funciones del proceso, no existiría ningún problema, ya que dichos datos son compensados o tratados en distintas partes del código.

2.2. Cámaras de tiempo de vuelo

Cuando se ha de obtener la descripción geométrica de una escena, el primer paso es estudiar los distintos métodos existentes y elegir el que mejor se adapte a las circunstancias. En la Figura 4 se puede observar una clasificación para enmarcar los sensores de tiempo de vuelo dentro de algunas de las tecnologías existentes para medir distancias.

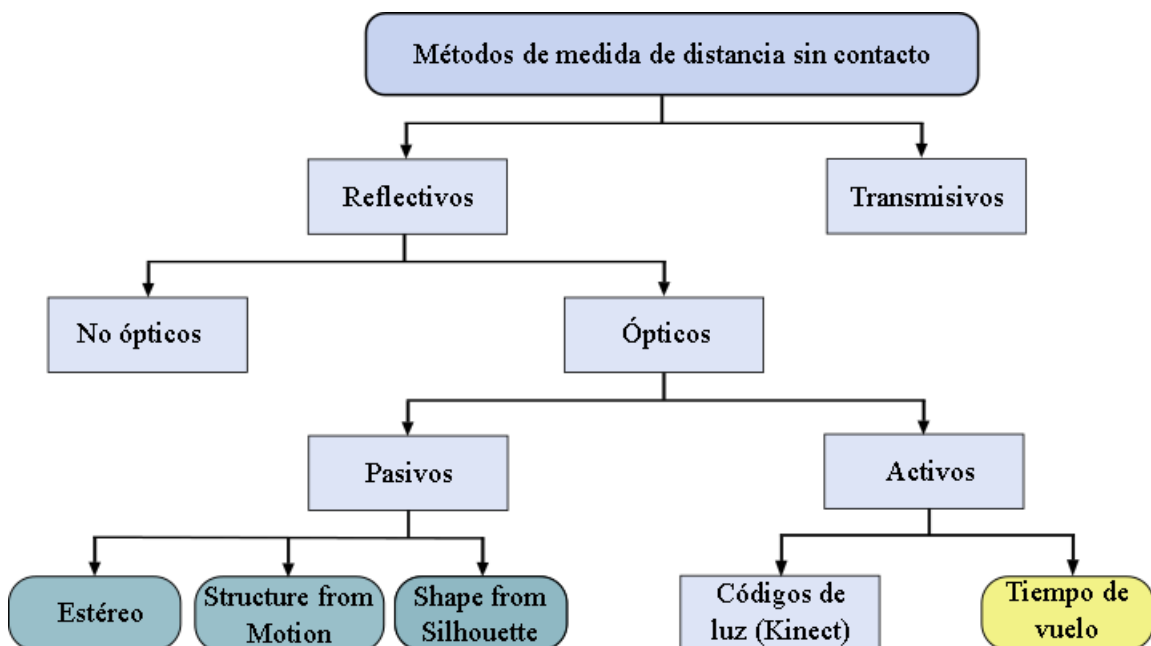


Figura 4. Distintos métodos de medida de distancia sin contacto

A pesar de que también existen técnicas de medida de distancia con contacto, solamente se van a estudiar aquellas que miden sin contacto. Dentro de ellas, hay técnicas transmisivas, es decir, que la radiación atraviesa la superficie a medir, como por ejemplo el escaneo mediante técnicas de tomografía y también técnicas reflexivas, en las que lo que se realiza es el análisis de la luz que se refleja en la superficie. Dentro de los métodos reflexivos hay algunos que no se basan en métodos ópticos, como puede ser medida por ultrasonidos o microondas, y otros que se basan en métodos ópticos. Los métodos ópticos, a su vez, pueden ser subdivididos en dos categorías: activos, que son aquellos que envían algún tipo de señal a la escena y capturan su reflexión en ella, y pasivos, que solamente reciben información sobre la posición del objeto mediante la luz que lo ilumine de manera natural. Dentro de los pasivos se encuentran métodos como la reconstrucción estéreo a partir del uso de dos cámaras, la reconstrucción a través del análisis del movimiento en la escena o la deducción de la distancia a partir de las siluetas o formas de la escena desde distintos puntos de vista. Por último, en la categoría de sensores

activos se encuentran los sistemas que proyectan códigos bidimensionales de luz conocidos, como la Kinect, y las cámaras de tiempo de vuelo.

Las cámaras de tiempo de vuelo realizan la medida de distancia sin contacto mediante un sensor activo óptico reflexivo. Básicamente, una cámara de tiempo de vuelo produce una imagen de profundidad en la cual cada píxel codifica la distancia al correspondiente punto de la escena, en lugar de indicar su color, como realizan las cámaras RGB. En la Figura 5 (derecha) se muestra un ejemplo de una imagen obtenida por una de las cámaras de tiempo de vuelo utilizadas en el trabajo actual, en la que la escala de color se corresponde con la distancia, con respecto a su imagen en pseudo escala de grises (izquierda).

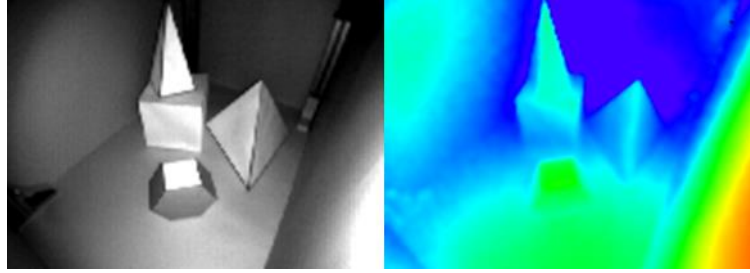


Figura 5. Izquierda: imagen en escala de grises. Derecha: imagen de distancias de una cámara ToF

Para comprender cómo se obtiene esta distancia, hay que estudiar el principio de medida de distancia mediante tiempo de vuelo, que aparece de forma gráfica en la Figura 6.

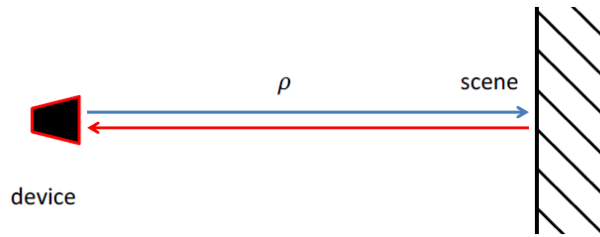


Figura 6. Principio de medidas ToF. Medida de tiempo.

Básicamente, consiste en calcular la distancia que hay entre el dispositivo o sensor hasta los elementos de la escena emitiendo luz y midiendo el tiempo que tarda en volver, ya que se conoce la velocidad c a la que viaja la luz ($c = 3 \cdot 10^8$ m/s). De esta manera, calculando el tiempo τ que tarda en volver la luz desde que se emite, se puede calcular la distancia d o profundidad mediante la ecuación (1):

$$d = \frac{c\tau}{2} \quad (1)$$

Sin embargo, este método de medida de tiempo tiene un problema, y dada la velocidad de la señal electromagnética, se necesitaría un sistema que midiera el tiempo con muchísima precisión, porque de lo contrario, se produciría un error elevado en distancia. Por ejemplo, si hubiera un error de 1 ns en la medida temporal, sustituyendo en la ecuación (1), obtendríamos un error de distancia de 3 m. Para asegurar esta precisión, han de diseñarse sistemas muy precisos en los que se pueda asegurar un error muy bajo, lo que implica un mayor coste económico. Por lo tanto, en las cámaras ToF se suele por medir la diferencia de fase $\Delta\phi$

2. Descripción del sistema

existente entre la señal infrarroja modulada emitida y la recibida, como se observa esquemáticamente en la Figura 7.

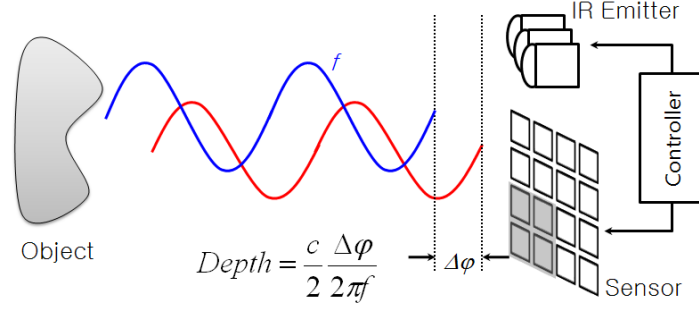


Figura 7. Principio de medidas ToF. Medida de desfase.

De esta manera, es más fácil y barato obtener una medida de distancia fiable. En la Figura 8 se observa una posible señal sinusoidal emitida (en azul), y la señal reflejada en la escena que se recibe (en rojo), observando la diferencia de fase entre ambas. Según este método, la profundidad o distancia d se calcula según la expresión (2), donde $\Delta\phi$ es el desfase medido entre ambas señales, c la velocidad de la luz y f_{mod} la frecuencia de la señal modulada que se emite.

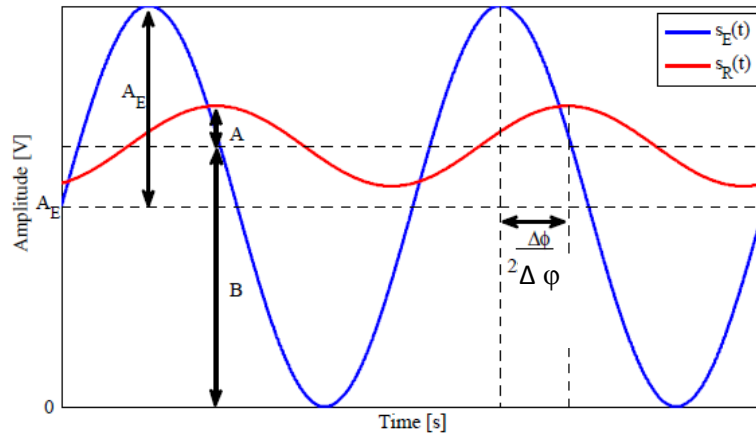


Figura 8. Señal emitida (azul) y señal recibida (rojo)

$$d = \frac{c}{2} \cdot \frac{\Delta\phi}{2\pi f_{mod}} \quad (2)$$

Por tanto, se ha de medir la diferencia de fase. Sin embargo, la fase no puede ser medida directamente, sino que se debe medir correlando [8] la señal óptica recibida en el sensor con la señal de referencia eléctrica, que está en fase con la luz modulada emitida, proceso que se lleva a cabo en la propia cámara ToF. Para medir la luz recibida, se utilizan unos elementos llamados 'smart pixels' [13] basados en la tecnología CMOS.

Por otro lado, la pseudo-imagen en escala de grises que proporciona la cámara es obtenida a partir de la amplitud de la señal recibida, que correspondería al valor de la amplitud A de la señal $s_R(t)$ de la Figura 8, y que como va a depender de la reflectividad de la superficie en la que se refleje la señal emitida $s_E(t)$.

Es importante tener en cuenta que las cámaras de tiempo de vuelo proporcionan una medida de distancias, no coordenadas tridimensionales, las cuales se pueden obtener

matemáticamente conociendo los parámetros intrínsecos de la cámara mediante un proceso de calibración.

2.3. Modelo Pinhole y parámetros de distorsión

Para poder reconstruir una escena en 3D es necesario conocer la correspondencia entre los píxeles de la imagen $(u, v)^T$ y las coordenadas del espacio $(X, Y, Z)^T$. Para ello, se va a utilizar el modelo pinhole, muy ampliamente conocido y utilizado desde hace bastantes años. No obstante, se incluyen las ecuaciones del modelo para hacer este trabajo autocontenido. Básicamente, describe de manera sencilla la relación matemática entre las coordenadas de un punto 3D y su proyección en un plano imagen ideal, como aparece en la Figura 9.

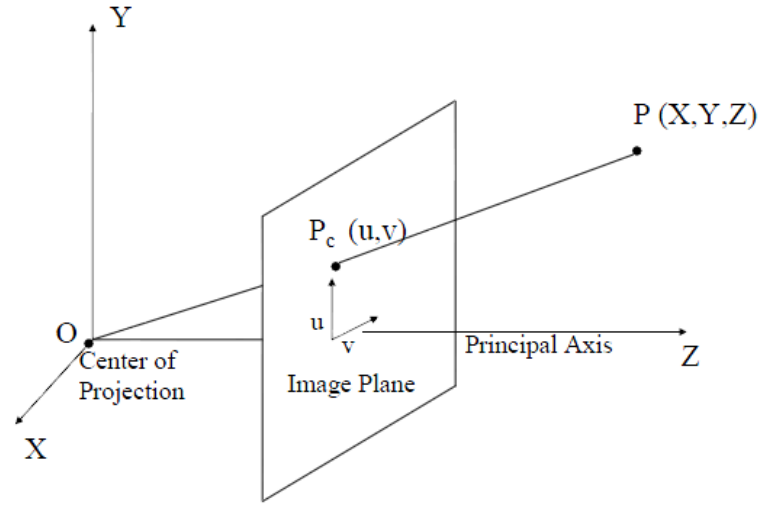


Figura 9. Modelo pin-hole de una cámara

Un punto 3D P de coordenadas $(X, Y, Z)^T$ se proyecta sobre el plano imagen en el punto 2D P_c , de coordenadas $(u, v)^T$, a través del rayo que une dicho punto P con el origen o centro de proyección O . Sabiendo que la distancia existente entre dicho centro de proyección y el plano imagen se conoce como distancia focal f , se pueden hallar las coordenadas de P_c mediante semejanza de triángulos:

$$\frac{f}{Z} = \frac{u}{X} = \frac{v}{Y} \rightarrow \begin{cases} u = f \frac{X}{Z} \\ v = f \frac{Y}{Z} \end{cases} \quad (3)$$

Esta relación también puede tratar de expresarse en coordenadas homogéneas, como se muestra en la ecuación (4), teniendo en cuenta que la distancia focal f puede variar si los factores de escala de los ejes X e Y del sistema no son iguales, obteniendo dos distancias focales distintas f_x y f_y .

2. Descripción del sistema

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \quad (4)$$

Por tanto, el punto P_c de la imagen se puede expresar según la ecuación (5):

$$P_c = (u, v, w)^T = \left(f_x \frac{X}{Z}, f_y \frac{Y}{Z}, 1 \right)^T \quad (5)$$

Sin embargo, es posible que el origen de coordenadas 2D del plano imagen no coincida con el punto de intersección del eje Z (eje principal o eje óptico) con el plano imagen, conocido como centro óptico. En dicho caso, se debe trasladar el punto P_c al origen en cuestión, lo cual queda expresado en la ecuación (6), teniendo en cuenta que c_x y c_y son las coordenadas del punto principal.

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = K \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \quad (6)$$

La matriz K se conoce como matriz de parámetros intrínsecos de la cámara. Esta matriz proporciona la conversión (escala y traslación) de la imagen proyectada de coordenadas del mundo real (m, cm...) a píxeles. Como se muestra en la Figura 10, los parámetros de esta matriz dependen directamente de la estructura del sensor de la cámara:

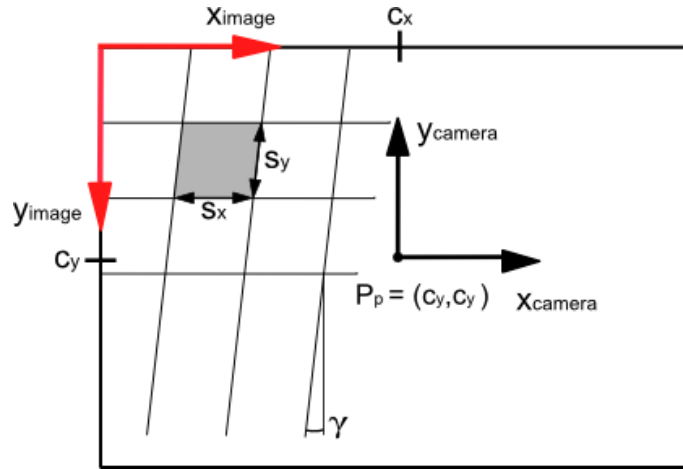


Figura 10. Correspondencia de parámetros de la matriz intrínseca con la estructura del sensor CMOS

- El parámetro (1, 2) de la matriz K se conoce como *skew* e indica la no ortogonalidad existente entre los ejes x e y de la cámara, lo que en la Figura 10 se correspondería con ángulo γ . Si ambos son prácticamente ortogonales, como ocurre en la mayoría de las cámaras de la actualidad, y también para las utilizadas en este proyecto, el *skew* puede ser aproximado a cero.
- También se observa en el propio sensor la posición del punto principal $P_p = (c_x, c_y)$, que es el punto donde el eje óptico intersecta al plano imagen.
- Por otro lado, si s_x es igual que s_y , el ratio entre el alto y el ancho de los píxeles sería la unidad, y por tanto, la distancia focal f_x sería igual a f_y , y habría una única distancia focal f en la matriz K . De no ser iguales, las distancias focales relacionadas con el tamaño del

píxel se pueden hallar según la ecuación (7). El significado es la distancia focal dividida por el ancho y el alto de un píxel respectivamente, y el resultado es la escala entre las coordenadas del mundo real (en el sistema métrico) y los píxeles de la imagen.

$$\begin{cases} f_x = f/s_x \\ f_y = f/s_y \end{cases} \quad (7)$$

Por último, si la cámara no tiene su centro de proyección en el punto tridimensional $O=(0, 0, 0)$ y está posicionada en alguna otra posición, o con una orientación distinta, es necesario recurrir a una rotación y una translación para conseguir este cambio de coordenadas, como se presenta en la ecuación (8), y de esta manera conseguir que el sistema de coordenadas de la cámara coincida con la configuración pinhole de la Figura 9.

$$\begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} = R_{\alpha,\beta,\gamma} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + T = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} \quad (8)$$

Esta operación aparece reflejada gráficamente en la Figura 11:

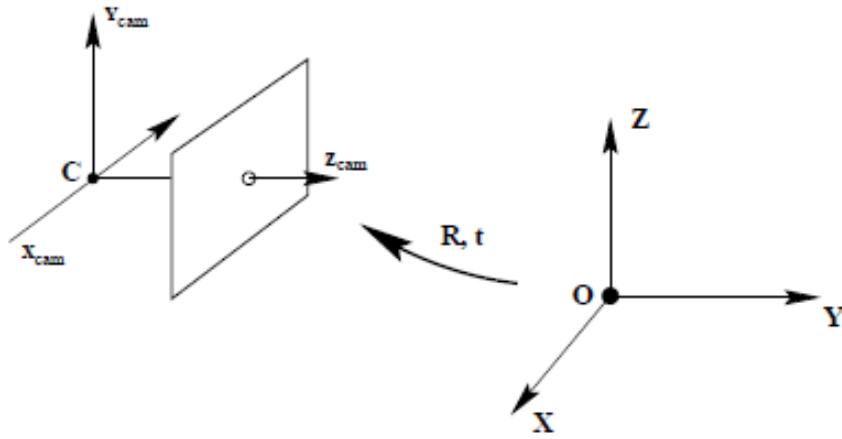


Figura 11. Rotación y translación aplicada

Por tanto, hallando estos parámetros, lo cual se consigue mediante el proceso de calibración, es posible modelar matemáticamente una cámara y conocer la relación existente entre los píxeles de las imágenes y el mundo real (correspondencia 2D-3D).

Sin embargo, el modelo pinhole descrito es bastante restrictivo ya que solamente se basa en la proyección de una escena a través de un punto. Pero en una cámara real esto no ocurre así, ya que dicho punto en realidad es un área de tamaño considerable cubierta por una lente, por lo que la imagen real va a presentar cierta distorsión y no va a ser como idealmente se esperaba según se ha definido mediante dicho modelo pinhole.

Se puede definir la distorsión como las aberraciones que se producen en la imagen debidas a la presencia de una lente. Principalmente, se suelen considerar dos tipos de distorsiones: radial y tangencial. Sin embargo, existen más tipos de aberraciones que no van a ser estudiadas ni modeladas en el presente proyecto por no considerarse que influya en el modelo significativamente.

2. Descripción del sistema

En primer lugar, la distorsión radial es la que sufre un rayo que pasa por el centro de la lente debido a la no linealidad de la Ley de Snell, como se muestra en la Figura 12. Si el sistema fuera ideal, el rayo que pasa por el centro de la lente no debería modificar su trayectoria, es decir, debería continuar propagándose sin ningún cambio de dirección. Sin embargo, en la realidad este rayo se desvía ligeramente, ocasionando las conocidas distorsiones de barril o de cojín, mostradas en la Figura 13.

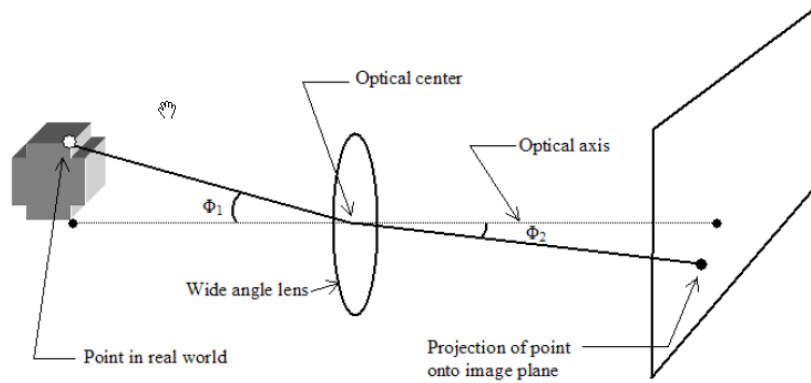


Figura 12. Efecto de la distorsión radial

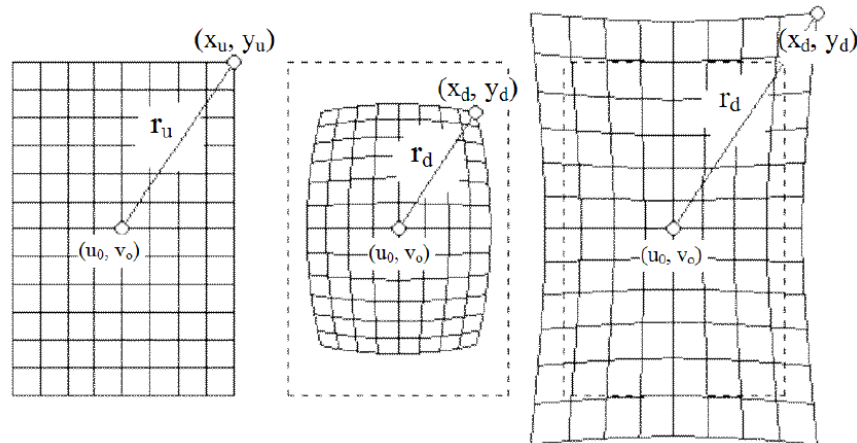


Figura 13. Efecto de la distorsión radial. Imagen sin distorsión (izquierda), con distorsión de barril (centro) y de cojín (derecha)

La distorsión tangencial es provocada por la desalineación existente en el sistema de lentes real de una cámara. Es decir, por la existencia de una desalineación entre el eje óptico en distintas partes del sistema. A pesar de que muchas cámaras actuales esta distorsión llega a ser casi despreciable, debido a que el acople mecánico de estas cámaras de bajo coste no es de gran calidad, va a ser muy importante tener en cuenta este parámetro. Estas distorsiones pueden ser modeladas por una serie de coeficientes, de manera que es posible corregir la distorsión de las imágenes, y a pesar de que en muchos casos esta distorsión es considerada nula, no es posible realizar esta aproximación en el proyecto actual, ya que si no se tiene en cuenta, las imágenes no se corrigen correctamente.

Por lo tanto, lo que se busca es conseguir una ecuación matemática que modele el efecto de la distorsión, para así poder corregirla y obtener una imagen no distorsionada. El modelo que se va a utilizar es conocido como modelo de fotogrametría y se muestra en la ecuación (9),

donde se pretende hallar las coordenadas de cada punto de la imagen sin distorsión $(x_u, y_u)^T$ a partir de las coordenadas de la imagen distorsionada $(x_d, y_d)^T$.

$$x_u = x_n(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + (p_1(r^2 + 2x_n^2) + 2p_2 x_n y_n) \quad (9)$$

$$y_u = y_n(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + (p_1(r^2 + 2y_n^2) + 2p_2 x_n y_n)$$

Donde $x_n = x_u - x_c$ y $y_n = y_u - y_c$, siendo $(x_c, y_c)^T$ el centro de distorsión, y r es la distancia radial desde el punto de la imagen con distorsión hasta el centro de distorsión, calculado como $r = \sqrt{x_n^2 + y_n^2}$. Los parámetros de distorsión que se tienen en cuenta en este modelo son $k_c = [k_1 \ k_2 \ k_3 \ p_1 \ p_2]$, de los cuales k_i corresponden a la distorsión radial y p_i a la tangencial.

2.4. Necesidad de calibrar una cámara ToF

Un sensor de tiempo de vuelo no devuelve las coordenadas tridimensionales de los objetos visibles en el campo de visión de la cámara. Lo que devuelve es la distancia radial existente entre dicho sensor y los puntos de la escena. Es decir, si observamos la Figura 14 y se tiene en cuenta que el punto número 1 corresponde con el origen de coordenadas de la cámara, y el punto número 2 es un punto a medir, las coordenadas tridimensionales de dicho punto serían:

$$P_2 = (a, b, c)^T \quad (10)$$

Sin embargo, la distancia del punto 2, que es el valor que devuelve una cámara ToF para cada píxel de la imagen de distancias correspondería a:

$$P_2 = \sqrt{a^2 + b^2 + c^2} = e \quad (11)$$

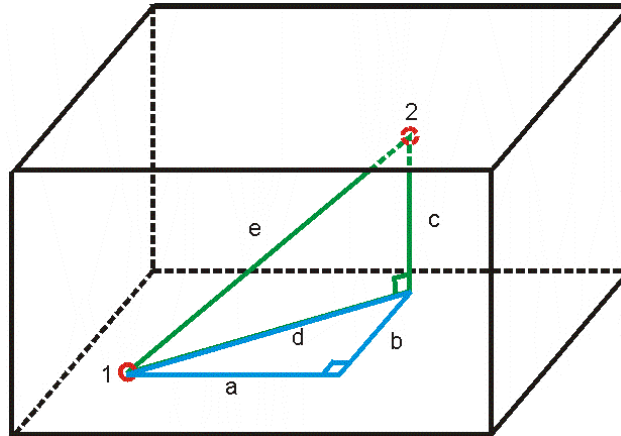


Figura 14. Coordenadas tridimensionales y distancia radial²

Si se pretende realizar un sistema de reconstrucción tridimensional, no es suficiente con conocer la distancia a los puntos de la escena, sino que se necesita conocer las coordenadas de cada uno de ellos, y para ello es indispensable conocer la matriz de calibración de la cámara y los parámetros de distorsión. No obstante, el fabricante proporciona una función que devuelve directamente las coordenadas tridimensionales de la escena a partir de unos parámetros iniciales de calibración genéricos para las cámaras cuando salen de fábrica. Sin embargo, debido a que la

² http://hlab.stanford.edu/~brian/euclidean_distance_in.html

posición de la lente se puede mover y que la calibración no está personalizada para ese sensor concreto sino para todos los del mismo tipo, se deben obtener datos más precisos de calibración. Por lo tanto, para llevar a cabo el presente trabajo se requiere calcular la posición 3D de cada punto a partir de los datos óptimos de calibración, y esto se puede conseguir de manera muy sencilla si se conoce la matriz de calibración intrínseca de la cámara llevando a cabo los siguientes pasos:

- En primer lugar, corregir la distorsión existente en la imagen acorde a los parámetros de distorsión obtenidos durante la calibración y aplicando la ecuación (9), obteniendo una imagen nueva sin distorsión.
- A continuación, calcular las coordenadas X e Y de cada punto a través de sus coordenadas en píxeles y los parámetros de calibración:

$$P' = \begin{pmatrix} X' \\ Y' \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}^{-1} \cdot \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \quad (12)$$

- En el paso anterior se ha asumido que los puntos estaban todos situados en una distancia $Z = 1$. En realidad no es así, sino que la cámara de ToF proporciona la distancia real. Por lo tanto, habría que multiplicar los puntos P' por su correspondiente distancia normalizada Z_n , y de esta manera obtener las coordenadas finales P :

$$Z_n = \frac{Z_{depth}}{\sqrt{(X')^2 + (Y')^2 + 1}} \quad P = Z_n \cdot P' \rightarrow \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = Z_p \cdot \begin{pmatrix} X' \\ Y' \\ 1 \end{pmatrix} \quad (13)$$

2.5. Descripción del sensor ToF y algoritmo de calibración

2.5.1. Selección de la cámara ToF

Una vez planteadas las bases del sistema y los objetivos que pretenden alcanzarse, hay que centrarse en los elementos que se van a utilizar para llevarlo a cabo, que en este caso son básicamente el elemento hardware (las cámaras de tiempo de vuelo) y el soporte software (MATLAB®).

En primer lugar, ha de elegirse un fabricante y modelo de cámara de tiempo de vuelo. Este tipo de cámaras son dispositivos cuyo precio en el mercado siempre ha sido muy elevado (superior a los 4000\$³) aunque están comenzado a surgir en la actualidad sensores de tiempo de vuelo de más bajo precio. Como se busca realizar un sistema con el menor coste posible, una primera opción podría ser utilizar la primera cámara Kinect de Microsoft [2], la cual es una de las cámaras comerciales que devuelve una medida de distancia con un precio más bajo. Sin embargo, a pesar de que esta cámara proporciona medida de profundidad, y también información RGB, no está basada en tiempo de vuelo, si no que calcula la distancia mediante la proyección de un patrón de puntos con luz infrarroja en la escena y el análisis de dicha proyección en ella. En la Figura 15 se puede observar un ejemplo de cómo se vería el patrón proyectado sobre una escena real, imagen a partir de la cual la cámara calcularía la profundidad.

³ <http://www.acroname.com/products/R317-SR4000-CW.html>



Figura 15. Patrón de la cámara Kinect proyectado sobre una escena

Por tanto, se decide buscar un sensor de tiempo de vuelo para probar su eficiencia, y además con el menor precio posible. Se elige la cámara CamBoard Nano⁴ del fabricante PMD Technologies, que se muestra en la Figura 16. En realidad, es un chip prototipo de muy bajo coste (alrededor de los 400€) en comparación con otros sensores de tiempo de vuelo existentes en el mercado, pero el fabricante deja muy claro que no es un modelo diseñado y testado para uso directo y comercial, sino que es un modelo diseñado para desarrolladores que quieran realizar aplicaciones de medida de profundidad a cortas distancias.

Esta cámara emite luz infrarroja (@820 nm) modulada a 30 MHz. Esta frecuencia es fija, no permite ser cambiada, por lo que se debe tener en cuenta que si se quiere utilizar más de una cámara de este modelo, no va a poderse obtener imagen de ellas simultáneamente, ya que se interferirían, habría que conmutarlas temporalmente. En este proyecto se van a utilizar 4 cámaras, pero no se pueden obtener datos de las 4 en el mismo instante temporal, sino que se encenderá el led de la cámara que se requiera en cada momento.



Figura 16. CamBoard Nano

Debido a que el método utilizado para medir la distancia es mediante diferencia de fase, solo se van a poder medir desfases entre 0 y 2π rad, ya que si se mide un desfase mayor que 2π rad, por ejemplo de 3π rad, sería confundido con π rad. Existen algoritmos que tratan de solucionar este problema (métodos de *phase unwrapping*), pero no van a ser tenidos en cuenta para este proyecto. Teniendo esto en cuenta, la distancia máxima que se puede medir de manera unívoca viene marcada por este aspecto y la frecuencia de modulación, según la expresión (15):

⁴ http://www.pmdtec.com/products_services/reference_design.php

$$d_{max} = \frac{c}{2 \cdot f_{mod}} = \frac{3 \cdot 10^8}{2 \cdot 30 \cdot 10^6} = 5 \text{ m} \quad (14)$$

Por lo tanto, el fabricante indica que la cámara puede medir distancias entre 20 cm y 5 m, y que distancias mayores serán confundidas. Sin embargo, se observa que realmente la cámara está optimizada para funcionar hasta 1 m, y que a partir de esa distancia, los valores de distancia medidos son en gran proporción incorrectos o inválidos.

Además, son unas cámaras de visión con un amplio campo de visión (FOV), de 90° x 68° y alcanzan un *frame rate* de hasta 90 fps. El hecho de que puedan adquirir hasta 90 frames por segundo indica que podrían utilizarse para trabajar con escenas en las que existe movimiento rápido. Sin embargo, hay que tener en cuenta que esto se ha calculado con el tiempo de adquisición más bajo posible, en el cual la calidad de la imagen mostrada no suele ser buena, y existen muchos píxeles con valor no válido. En la práctica, ha de subirse el tiempo de adquisición, y además es conveniente realizar un mediado de varias imágenes a lo largo del tiempo, de manera que las imágenes obtenidas, tanto de amplitud como de medida de profundidad, tengan una cierta calidad que pueda ser aceptable. Por lo tanto, el *frame rate* efectivo es menor de 90 fps.

Por otro lado, las cámaras elegidas tienen una resolución de 120x160 píxeles, y como consecuencia de ello, será difícil obtener imágenes gran detalle de la escena. Además, esta cámara no tiene sensor de imagen RGB ni en escala de grises, si no que utiliza la radiación infrarroja que emite para el cálculo de profundidad para proporcionar una imagen de amplitudes en tonos de grises que represente la escena. Sin embargo, las diferentes intensidades que presenten las partes de la escena capturada no se deben solamente al color que presenten los distintos objetos, sino a la reflectividad que presenten dichas superficies ante la luz modulada infrarroja emitida.

Debido a ello y a la baja calidad del sensor (válido, sin embargo, para aplicaciones sencillas en las que se deba medir distancia), existen problemas en la calibración. En primer lugar, el fabricante vende las cámaras diciendo que han sido calibradas individualmente y proporciona sus parámetros de distorsión y de la matriz intrínseca. Sin embargo, estos parámetros son dados de manera general y no individual para cada cámara, y además, las cámaras vienen con una lente que se puede enfocar, por lo que los parámetros de calibración no pueden ser previamente calculados independientemente del enfoque de la lente. Por esto, es necesario calibrar para lograr imágenes con la mayor precisión posible.

2.5.2. Calibración

Para ello, primero hay que obtener imágenes de amplitud con un patrón de medidas conocidas en distintas posiciones, y en cada imagen hallar con exactitud los puntos característicos en coordenadas pixélicas, con el objetivo de pasarlos como base al algoritmo de calibración. Dado que la resolución es baja, dichos puntos característicos han de ser calculados con mucha más precisión de lo que podría ser necesario en sensores con más píxeles, ya que un solo píxel de error de cálculo equivaldría a un error demasiado alto en las coordenadas del mundo. De hecho, es necesario que la precisión de cálculo sea subpixélica (menor que 1 píxel). Por lo tanto, deben utilizarse patrones con figuras grandes, por lo que el número de puntos característicos va a disminuir. Además, las imágenes que se obtienen pueden presentar grandes

reflexiones en determinados objetos dependiendo del material del que estén elaborados, por lo que hay que buscar materiales que sean mate frente a la radiación de la cámara.

Las cámaras de tiempo de vuelo presentan un error conocido por ‘multi-camino’, y se debe a que un rayo proyectado en la escena puede sufrir varias reflexiones, y volver entonces al sensor. El modelo de medida de estas cámaras está preparado para medir únicamente el rayo directo, y no evitan los efectos que provienen de reflexiones múltiples. Esto produce un error en la medida que no va a ser modelado ni corregido en este trabajo, pero ha de tenerse en cuenta que determinadas zonas (esquinas redondeadas, agujeros...) pueden aparecer distorsionadas con respecto a la realidad debido a este motivo.

Por otro lado, aparte de la problemática de las cámaras, las funciones de calibración que proporciona MATLAB® son muy utilizadas y aceptadas como válidas cuando se ha de realizar una etapa de calibración en cualquier proyecto de visión. Sin embargo, realmente han sido utilizadas con cámaras con una resolución más alta de la de estos sensores, y se ha comprobado que presentan dificultades y deficiencias para calibrar cámaras de este tipo.

Más adelante se detallan, explican y justifican cada uno de los problemas que se acaban de introducir, junto con los métodos adoptados para solventarlos.

3. Extracción de puntos característicos para la calibración de la cámara ToF

Como ya se ha explicado previamente, el objetivo de la calibración es determinar el mapeado o la correspondencia existente entre la escena del mundo real y la imagen. Relacionando puntos de la escena de posición relativa conocida con su equivalente en la imagen, el modelo de la cámara propuesto puede ser calculado. Esta correspondencia de puntos junto con cierta información a priori sobre los parámetros internos de la cámara forman la base de la calibración.

Por lo tanto, para calibrar, hay que elaborar un patrón que tenga unos puntos característicos que sean conocidos con precisión, y tomar imágenes de dicho patrón colocado en distintas posiciones de la escena, detectando en todas las imágenes dichos puntos característicos. Con este proceso, ya se obtendrían una serie de correspondencias entre puntos del mundo real y su proyección en la imagen, lo que sirve de base para la calibración. El problema en el caso de las ToF utilizadas es que debido a su baja resolución se complica en gran medida la detección de estos puntos, por lo que hay que diseñar un patrón y un algoritmo de búsqueda que satisfaga el objetivo. En la Figura 17 se muestra el patrón que va a ser utilizado (izquierda) y una imagen ejemplo capturada en la escena (derecha).

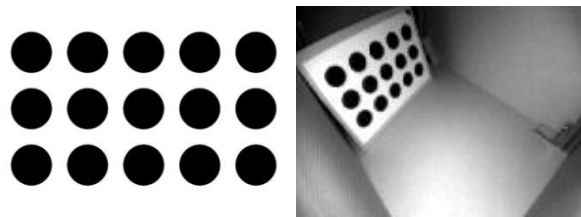


Figura 17. Patrón utilizado para la calibración

El punto de partida para la calibración de las cámaras de tiempo de vuelo utilizadas en este proyecto es el análisis de las pseudo-imágenes de escala de grises que proporciona el sensor. Además, se va a utilizar un patrón bidimensional, es decir, un patrón impreso en un plano. Debido a la naturaleza de las cámaras, la superficie sobre la que esté impreso el patrón debe ser mate, es decir, no debe presentar reflexiones al impactar la radiación infrarroja modulada sobre él, porque esto produciría brillos en la imagen, dando lugar a posibles imprecisiones en la detección de los puntos de control, como por ejemplo se puede observar que ocurre en la Figura 18. Además, este patrón impreso debe ser colocado sobre una superficie que sea completamente rígida.

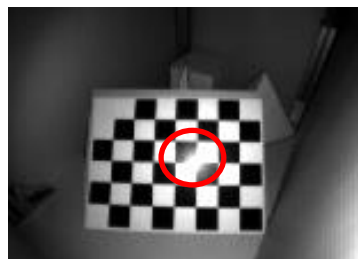


Figura 18. Imagen de un patrón en el que se producen reflexiones

A pesar de que es muy fácil obtener un patrón de calibración cuyas posiciones de los puntos de calibración sean perfectamente conocidas sobre dicho plano, la detección de dichos puntos en las imágenes obtenidas no es en absoluto tan trivial, y esto tiene especial importancia en el proyecto actual debido a la baja resolución de las cámaras. En primer lugar, hay que tener en cuenta que los puntos de calibración han de estar bastante separados entre sí y que tengan un gran tamaño para poder distinguirlos con claridad en la imagen, y esto hace que el número de puntos que se pueden colocar en el patrón sea menor que en una cámara de alta resolución.

Por otro lado, esta baja resolución provoca que exista una imprecisión en la medida que aumenta cuanto más lejos se encuentre el punto que está siendo capturado. Para conocer la magnitud de esta falta de precisión, sabiendo que la distancia focal en mm de la cámara es de unos 4.3 mm, que el sensor tiene unas dimensiones de 7.2 mm x 5.4 mm y que las imágenes tienen una resolución de 160 x 120 píxeles, se puede calcular primero el ancho de cada píxel físico mediante la ecuación (15), y a partir de ella se calcula la incertidumbre a una distancia determinada (por ejemplo, a 1 metro), según se desarrolla en la expresión (16).

$$\text{ancho}_{\text{pixel}} = \frac{\text{ancho}_{\text{sensor}}}{n_{\text{píxeles}}} = \frac{0.0072}{160} = 45 \mu\text{m} \quad (15)$$

$$\text{incertidumbre}_{|1 \text{ m}} = \frac{\text{ancho}_{\text{pixel}} \cdot 1 \text{ m}}{f} = \frac{45 \cdot 1}{0.0043} = 1.05 \text{ cm} \quad (16)$$

La incertidumbre calculada se refiere solamente en el eje X. Es decir, que de la coordenada x obtenida de un objeto situado a un metro, habrá una incertidumbre de ± 1.05 cm. A esto habría que añadirle la imprecisión calculada en el eje Y, más la imprecisión en la medida de la distancia asociada a ese punto, por lo que finalmente existe una incertidumbre mayor. Además, esta imprecisión es únicamente debida al efecto producido por la baja resolución de las cámaras, es decir, no se ha tenido en cuenta ningún otro parámetro que hace que la proyección real sobre el plano imagen difiera de la ideal, como por ejemplo, la distorsión. Sin embargo, al contrario que el error producido por la distorsión, que como ya se ha explicado en la sección 2.3, puede ser corregido tratando la imagen, la incertidumbre producida por la baja resolución no puede ser corregida, porque no es un error que tenga una magnitud fija, sino que solamente se pueden establecer unos márgenes entre en los que se podría encontrar dicho error. A mayor resolución, menor es esta incertidumbre a igual distancia.

Esto indica que es muy importante detectar los puntos de control en las imágenes con la mayor precisión posible con el fin de no añadir ningún error más al resultado, ya que la calibración nunca podrá ser mejor que los datos a partir de los cuales ésta se calcula, y para reconstruir en 3D es muy importante partir de unos datos de calibración rigurosos de la cámara, y si se quieren unir los datos de varias vistas, también es esencial conocer con mucha exactitud las matrices de rotación y traslación. Existen varias maneras de mejorar la precisión de la extracción de estos puntos, como por ejemplo, ayudándose de una transformación proyectiva [6] para eliminar posibles errores de la lente o de las matrices proyectivas entre cámara e imagen, tratando de eliminar la distorsión y volviendo a proyectar la escena sobre el plano imagen para hallar los puntos con más exactitud [7], o mediante el cálculo de dichos puntos de control con una medida abstracta menor a un píxel, lo que se conoce como precisión subpíxelica.

3.1. Precisión sub-pixélica

La estimación sub-pixélica [3] es el proceso de estimar el valor de una cantidad geométrica con una precisión menor que un píxel, aunque los datos originales fueran originalmente tomados en un espacio cuantificado por una unidad entera de píxel. En la Figura 19 se muestra gráficamente un ejemplo de cómo aproximar una serie de píxeles de una imagen que forman parte del contorno de un círculo a una curva (en azul), con precisión sub-pixélica.

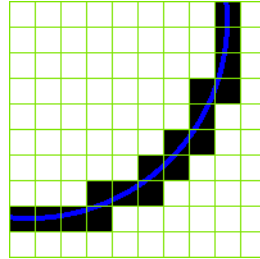


Figura 19. Estimación sub-pixélica de una curva

En muchas ocasiones, cuando los datos continuos son muestreados o cuantificados en unidades de 1 píxel, se suele asumir que la información a una escala menor que la unidad de muestreo se pierde. Sin embargo, es posible estimar cantidades por debajo de esta unidad. Las bases necesarias para esta estimación pueden ser:

- Modelos de variación espacial predecibles o esperados. Por ejemplo, las líneas o los bordes de una imagen producen patrones característicos conocidos sobre los datos cuando son medidos, permitiendo ajustar los datos obtenidos al modelo ideal, como por ejemplo, sabiendo que es una línea recta o una circunferencia (Figura 19).
- Integración en el espacio durante el muestreo. Los sensores que obtienen imagen suelen integrar una señal continua a lo largo del espacio (o del tiempo), haciendo que dichos valores obtenidos finalmente dependan de la posición relativa en la que se encuentran en la ventana de muestreo.
- El conocimiento de la función de dispersión de punto (PSF), que describe la respuesta del sensor a una fuente de un solo punto. Sería como una especie de respuesta impulsiva. En la Figura 20 se muestra cómo resultaría la imagen de un objeto ideal (izquierda, arriba) si se conoce que la PSF de la cámara es la que se muestra a la izquierda y abajo. Como se observa, es posible basarse en esta información para disminuir la borrosidad de una imagen.

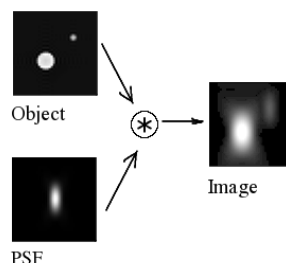


Figura 20. Resultado de aplicar la PSF al objeto inicial ideal. ⁵

⁵ http://es.wikipedia.org/wiki/Funci%C3%B3n_de_dispersi%C3%B3n_de_punto

Basándose en estos conceptos, podrían utilizarse una de las cuatro aproximaciones que se indican a continuación con el fin de alcanzar precisión sub-pixélica:

- Interpolación. Un ejemplo de ello sería lo que se muestra en la Figura 21, en la que aparece lo que podría ser una estimación sub-pixélica de un borde en una imagen, que es una recta. Otra aproximación de este estilo podría ser interpolar una curva o una superficie encontrando la posición óptima sobre la curva reconstruida (usando operaciones de correlación, por ejemplo).
- Integración. Correspondería, por ejemplo, a lo que se va a realizar en ese trabajo, que sería la estimación del centroide de un punto circular o elíptico. Para ello, se asume que las desviaciones de los píxeles pertenecientes a los bordes del círculo junto con los niveles de gris de la ventana de búsqueda tenida en cuenta pueden ser acumulados para proporcionar una estimación más robusta de la posición del centroide.
- Aproximación mediante serie de Taylor.
- Correlación de fase.

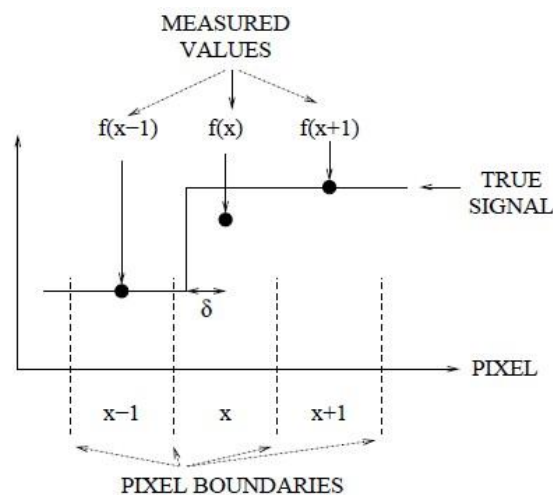


Figura 21. El valor del píxel es hallado mediante interpolación con los píxeles próximos.

3.2. Elección del patrón

Por tanto, el primer aspecto a determinar es el patrón que va a ser utilizado para capturar imágenes válidas sobre las que determinar la posición de los puntos de control, cuyas posiciones relativas entre sí en unidades métricas sea perfectamente conocida. Como ya se ha indicado, se va a elegir un patrón en dos dimensiones con un motivo impreso sobre él y colocado sobre una superficie totalmente rígida y sin ninguna rugosidad. Además, es importante elegir un patrón que esté hecho de un material que no presente altas reflexiones ante la radiación de la cámara de tiempo de vuelo para que no ocurra lo mostrado en la Figura 18.

El siguiente aspecto a determinar es cómo van a ser los puntos característicos a detectar. Comúnmente, en temas de calibración se suele utilizar un tablero de ajedrez, y los puntos de control que han de buscarse sobre este patrón son las esquinas que forman sus casillas, como aparece en la Figura 22. En esta figura se muestra un patrón ejemplo con las esquinas que se deberían encontrar marcadas con cruces rojas.

3. Extracción de puntos característicos para la calibración de la cámara ToF

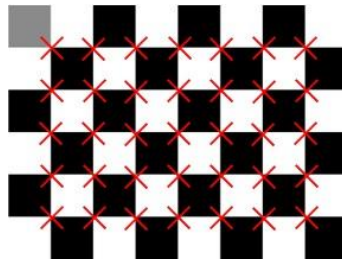


Figura 22. Patrón de tablero de ajedrez para detectar esquinas

Por lo tanto, se imprime este patrón y se toman imágenes de él en varias posiciones de la escena, tal y como aparece en la Figura 23.

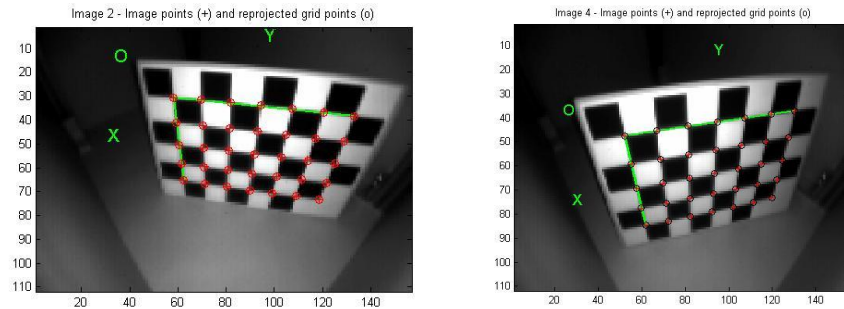


Figura 23. Búsqueda de esquinas con MATLAB®

En primer lugar, se decide utilizar las funciones que proporciona la toolbox de MATLAB® para la búsqueda de esquinas⁶, y se observa que para que las encuentre, es necesario colocar el patrón cerca de la cámara y en una posición en la que la inclinación del patrón no sea excesivamente alta. Por lo tanto, con esta búsqueda de puntos característicos no se pueden cubrir correctamente todas las zonas de la escena en varias posiciones del patrón, porque en dichos casos el algoritmo no encuentra las esquinas. Dos ejemplos de ello aparecen en la Figura 24.



Figura 24. Posiciones del damero en las que no se ha podido encontrar las esquinas.

Como se ha indicado, es necesario hallar la posición de los puntos característicos, las esquinas, con precisión sub-pixélica. Una manera de realizar esto es mediante interpolación. Una línea puede ser ajustada a los puntos de una imagen que se consideran co-lineales, y de esta manera se pueden calcular las líneas que definen los lados de los cuadros o casillas del damero y formar una rejilla. Las esquinas serían las intersecciones de dichas rectas calculadas. Este es un método ampliamente utilizado, pero sin embargo, como se ha observado en los ejemplos, con una cámara de tiempo de vuelo de baja resolución es muy difícil determinar los puntos que pertenecen a los bordes con precisión, y para que funcionase bien este método, el patrón de calibración tendría que estar muy cerca de la cámara. Por tanto, si se aleja el patrón, apenas se podrían detectar esquinas, que es lo que se ha observado. Sin embargo, en el apartado de calibración se analizarán los datos que proporcionan estos patrones para demostrar de manera numérica que los resultados obtenidos no son satisfactorios.

⁶ http://www.vision.caltech.edu/bouguetj/calib_doc/htmls/example.html

Como segunda opción en la búsqueda de un patrón válido, se planteó tratar de buscar las esquinas mediante la misma librería de calibración que utiliza MATLAB®, pero implementada mediante la librería OpenCV⁷. Sin embargo, los resultados de búsqueda de puntos de control eran aún peores, ya que en la mayoría de las imágenes no encontraba siquiera las esquinas, por lo que se abandonó esta rama de investigación y se decidió continuar buscando soluciones válidas mediante la herramienta de MATLAB®. De todas maneras también se van a comparar resultados obtenidos con este método, pero se han podido hallar con un set de imágenes muy reducido en las cuales el patrón está colocado en posiciones muy cercanas a la cámara, y totalmente enfrentado al sensor, sin apenas inclinación, ya que si se alejaba el patrón o se inclinaba, el algoritmo de OpenCV no detecta los puntos característicos.

A continuación, se pensó en utilizar un método de interpolación para realizar un zoom en las imágenes y tratar de encontrar las esquinas de esta manera. Efectivamente, MATLAB® encontraba en prácticamente todas las imágenes, o al menos aumentaba notablemente el ratio de imágenes detectadas frente a no detectadas, pero debido a la poca información al realizar la interpolación, éstas eran imágenes borrosas en las que se perdía en mayor medida la posición real del punto de control, consiguiendo un aumento de la incertidumbre en la medida, y en consecuencia, alejándose bastante de la aproximación sub-pixélica que es totalmente esencial obtener.

La siguiente opción que intentó fue disminuir el número de cuadros que existen en el patrón, aumentando así el tamaño de los cuadros que forman el damero, tratando de que así se calcule la posición de dichos puntos con más precisión. El objetivo que se persigue en este caso es el mismo que al hacer zoom en el planteamiento anterior. A pesar de que al realizar la calibración de una cámara es recomendable que existan un alto número de puntos característicos localizados en cada imagen para conseguir más información y poder mejorar los datos durante el proceso iterativo, si se aumenta el número de puntos, disminuye el lado de los cuadrados del damero, lo que es perjudicial para cámaras de baja resolución. Por lo tanto, se trata de buscar una solución de compromiso entre el número de puntos y tratar de hallar la mejor precisión posible. Para ello, se probó con patrones como los de la Figura 25. El patrón de la derecha tenía un número menor de cuadros que el requerido para la función de encontrar esquinas de MATLAB®, por lo que no pudo ser utilizado. El patrón de la izquierda introdujo ciertas mejoras, pero aun así no se obtenía la precisión deseada en la obtención de puntos característicos ni en la calibración.

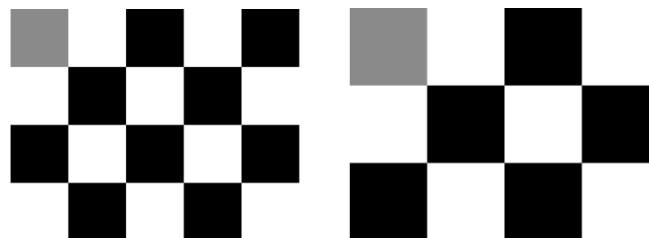


Figura 25. Patrones damero de mayor tamaño de cuadros

Por lo tanto, se tomó la decisión de abandonar el método de búsqueda de esquinas de MATLAB® y buscar otro tipo de patrón con otras características que diera mejor resultado con cámaras de bajo resolución, y diseñar un programa que permitiera su detección. Basándose

⁷ http://docs.opencv.org/doc/tutorials/calib3d/camera_calibration/camera_calibration.html

3. Extracción de puntos característicos para la calibración de la cámara ToF

también en la misma idea de utilizar o integrar la información de muchos puntos para hallar uno en concreto (el de control), se puede hallar con precisión sub-pixélica el centro de gravedad de una elipse proyectada. Esto se puede aplicar para generar un patrón con círculos cuyos centros tengan posiciones relativas entre sí conocidas en dicho patrón, tomar varias imágenes de éste en distintas localizaciones y orientaciones de la escena en la que se trabaja, y hallando el centro de gravedad de las elipses proyectadas que resultarán como la media ponderada de sus coordenadas y escala de grises. Tras probar con diferentes tamaños de círculos, distancias entre centros y posiciones relativas entre ellos, como se observa en la Figura 26, finalmente se escoge utilizar el patrón de la derecha de dicha figura, que es un patrón de 3x5 circunferencias impreso sobre una cartulina mate de color blanco tamaño A3, como se había supuesto desde un principio. Este patrón funciona bien porque son círculos bastante grandes, y de esta manera se tiene más información para localizar el centroide, y a su vez están suficientemente separados para no ser confundidos si el patrón se sitúa lejos de la cámara. A la vez, el número de puntos de control es bastante alto, ya que al haber 3 filas de 5 circunferencias, hay 15 centroides que van a buscarse en cada imagen.

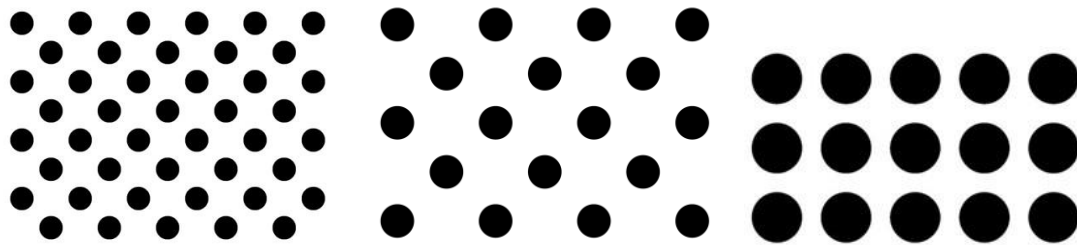


Figura 26. Patrones de calibración circulares. Patrón elegido (derecha)

Además, para encontrar correctamente el centroide de este patrón se ha decidido diseñar una función propia mediante scripts de MATLAB® que se encargue del tratamiento de la imagen capturada, su acondicionamiento y la búsqueda de los puntos de control con precisión sub-pixélica, como se explica a continuación en el apartado 3.3.

3.3. Búsqueda de las elipses y cálculo de los centroides

Una vez elegido un patrón que proporciona buenos resultados, han de tomarse varias imágenes (con cada cámara de manera independiente, en primer lugar) con el patrón situado en varias posiciones significativas de su campo de visión, tratando de abarcar todas las zonas de la imagen posibles. Es importante tomar imágenes en las que se sitúen puntos característicos en las zonas de los bordes de la imagen, que es donde la distorsión es mayor, y se necesita información de esas áreas para poder corregirla. A manera de ejemplo, se muestran en la Figura 27 varias imágenes tomadas con una misma cámara y el patrón situado en distintas posiciones.



Figura 27. Distintas imágenes tomadas con el patrón en distintas posiciones

Por lo tanto, el siguiente paso es la búsqueda de los círculos existentes en el patrón, o mejor dicho, las elipses proyectadas que resultan, para una vez encontradas poder calcular el centroide, que serán los puntos de control necesarios para el proceso de calibración. En primer lugar se trata de utilizar alguna de las funciones disponibles para ello en MATLAB®, así como la de OpenCV, pero ninguna de ellas da buenos resultados, ya que no encuentran los centroides en la mayoría de las imágenes. Esto ocurre según se esperaba, ya que como se ve en las imágenes de la Figura 27, las elipses detectadas tienen un tamaño muy pequeño, y en algunos casos, se pueden confundir entre sí al estar muy próximas. Además, en muchos de los casos los píxeles de borde no son muy precisos (debido a la baja resolución), y un solo píxel de error en el cálculo del centroide empeoraría los resultados en gran medida. Por lo tanto, con el propósito de detectar correctamente las elipses proyectadas se ha desarrollado un script en que realiza las funciones esquematizadas en la Figura 28.

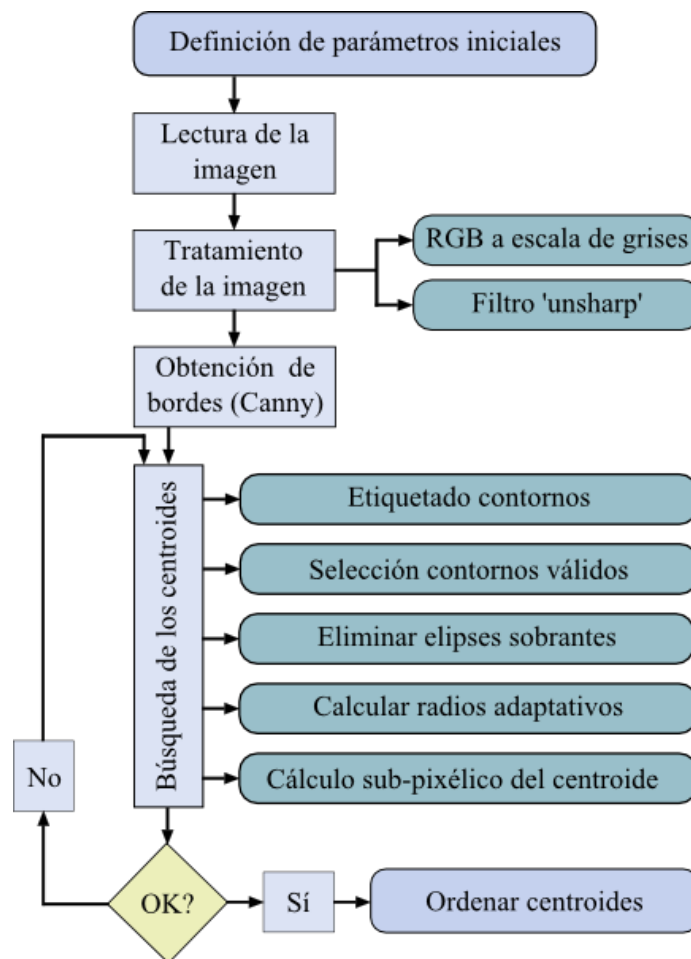


Figura 28. Flujograma de la búsqueda de elipses y cálculo de posición de los centroides

En primer lugar, se definen los valores iniciales de las restricciones que deben cumplir los distintos elementos que se encuentran en las imágenes para que puedan ser considerados elipses pertenecientes al patrón: tamaño máximo, mínimo, excentricidad, contorno cerrado... Inicialmente se programa una estimación de dichos parámetros, y empíricamente se ajustarán a un valor válido para todos los casos. A continuación, se lee la imagen almacenada, se convierte a escala de grises y se le aplica un filtro que facilite la detección de contornos. Se aplica otro filtro para detectar los bordes de las imágenes mejoradas, obteniendo todos los contornos que aparezcan. Entonces comienza un bucle en el que se etiquetan los contornos obtenidos y solamente se seleccionan aquellos que cumplen con las restricciones iniciales. Como se conoce

3. Extracción de puntos característicos para la calibración de la cámara ToF

que el patrón tiene 15 círculos, se programa una función de seguridad que trata de eliminar círculos no pertenecientes al patrón en caso de que se hayan detectado más de 15 candidatos. El siguiente paso es el cálculo de un radio adaptativo en torno al centroide del objeto encontrado en función del tamaño del círculo y la distancia a sus círculos contiguos para, finalmente, pasárselo como parámetro a la función que se encarga de calcular de manera sub-pixélica el centroide de las elipses encontradas.

Por último, se muestran los centroides calculados a través de la función que dota al sistema de una precisión por debajo del píxel y se da la opción de validarlos y pasar al siguiente paso, que es la ordenación de dichos puntos de control (sección 3.4), o si, por el contrario, se considera que los centroides no están correctamente calculados y se quieren realizar ciertas modificaciones con la intervención o supervisión del usuario, con el objetivo de mejorar la precisión del cálculo y obtener datos de calibración lo más fiables posibles.

En las siguientes secciones se detallan cada una de las etapas brevemente explicadas, terminando con la consecución del objetivo final, que es encontrar con éxito y precisión los centroides en las imágenes, pudiéndolos almacenar en un array ordenado siempre de la misma manera que sirve como datos iniciales para el proceso de calibración de la cámara.

3.3.1. Filtro ‘unsharp’

En primer lugar, se lee de directorio la imagen de amplitudes obtenida de la cámara y se pasa a formato ‘*gray scale*’ (escala de grises) de MATLAB®, el cual asigna a cada píxel de la imagen un valor entero entre 0 (negro) y 255 (blanco) correspondiente o proporcional al nivel de gris.

Para detectar los bordes, en el siguiente paso sería ideal que las imágenes fueran lo menos borrosas posibles y tuvieran un gran contraste en las zonas de cambio donde se quieren detectar bordes, teniendo una gran definición de estas características. Inicialmente, esto ya se favorece con el uso de un patrón de máximo contraste: fondo blanco y círculos negros. Sin embargo, este contraste disminuye al tomar la imagen ya que los todos los sensores, en mayor o menor medida, proporcionan una imagen con un nivel de borrosidad. Además, la baja resolución del sensor tampoco ayuda en esta tarea. Por tanto, con la idea de mejorar la imagen y obtener mejor los bordes, se va a filtrar la imagen con una máscara ‘*unsharp*’. Para entender gráficamente qué hace este filtro, se muestra su efecto sobre la imagen de una espiral en blanco y negro en la Figura 29.



Figura 29. Filtro unsharp. Imagen inicial (arriba) e imagen tras la aplicación del filtro (abajo)⁸

⁸ <http://en.wikipedia.org/wiki/File:Usm-unsharp-mask.png>

Básicamente se podría decir que una máscara '*unsharp*' exagera la diferencia de brillo a lo largo de los bordes de una imagen, consiguiendo que los bordes aparezcan más pronunciados, es decir, que la transición de niveles de gris en torno a los píxeles de los bordes sea más abrupta, como se muestra detalladamente y con zoom en la Figura 30.

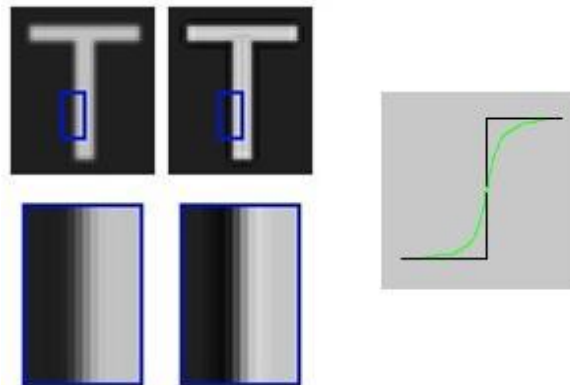


Figura 30. Bordes en la imagen inicial (izquierda), tras aplicar el filtro *unsharp* (centro) y transición en los bordes (real en verde e ideal en negro)⁹

El filtro '*unsharp*' es un filtro paso alto de segundo orden que detecta los cambios de alta frecuencia y los amplifica, obteniendo una imagen más nítida. En MATLAB® se realiza aplicando una máscara Laplaciana negativa de dimensiones 3x3 del tipo de las mostradas como ejemplo en las matrices (17).

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} -0.4 & -0.2 & -0.4 \\ -0.2 & 3 & -0.2 \\ -0.4 & -0.2 & -0.4 \end{bmatrix} \begin{bmatrix} -1/6 & -2/3 & -1/6 \\ -2/3 & 13/3 & -2/3 \\ -1/6 & -2/3 & -1/6 \end{bmatrix} \quad (17)$$

Por tanto, mediante la función *imfilter*, MATLAB® filtra la imagen de entrada en escala de grises con una determinada máscara tipo *unsharp* generada mediante la función *fspecial*, obteniendo finalmente la imagen filtrada mediante un proceso correlación (puede elegirse realizarlo mediante convolución, si se requiere). En la Figura 31 se muestra el efecto de la aplicación de esta máscara a una imagen real capturada con el patrón. Se observa cómo aumenta el contraste entre los círculos negros y el fondo del patrón blanco tras aplicar el filtro, haciendo que sea más fácil encontrar los bordes en la segunda imagen. Sin embargo, también se observa que la calidad final que se consigue para la detección de los círculos a utilizar para el cálculo de sus centroides es deficiente, lo que dificulta el proceso y es motivo por el que no se ha conseguido utilizar otras funciones previamente diseñadas para cumplir estos objetivos.

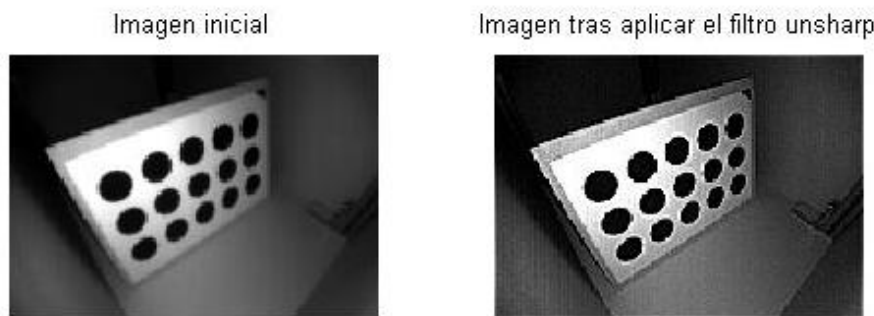


Figura 31. Imagen del patrón inicial (izquierda) y tras aplicar el filtro *unsharp* (derecha)

⁹ <http://www.cambridgeincolour.com/tutorials/image-sharpening.htm>

3. Extracción de puntos característicos para la calibración de la cámara ToF

3.3.2. Detector de bordes 'canny'

La detección de bordes es una operación matemática realizada en visión que consiste en identificar los puntos de una imagen en los que se producen discontinuidades en la intensidad, o mejor dicho, cambios de intensidad abruptos [11]. Encontrar los bordes servirá para definir los contornos de los que se compone la imagen y poder dividirla en partes y estudiar las características de cada una de ellas, pudiendo seleccionar las que corresponden a las elipses del patrón.

De entre todas las maneras disponibles de hacerlo, se va a utilizar la función 'edge' de MATLAB®, y se van a contrastar los métodos Sobel, Prewitt, Roberts, Canny, Laplaciana y Laplaciana de la Gaussiana (para más información, ANEXO I- Detectores de bordes) para comprobar cuál de ellos proporciona mejores resultados. Con ello, se observa que cuando el patrón se encuentra enfrenteado a la cámara y a una distancia no muy alejada, todos los métodos detectan correctamente las 15 elipses del patrón, como se observa en la Figura 32. Sin embargo, cuando el patrón está lejos y con un ángulo de rotación con respecto a la cámara alto, es cuando es más complicado detectar bordes, básicamente porque disminuye el número de píxeles que separan las elipses entre sí.

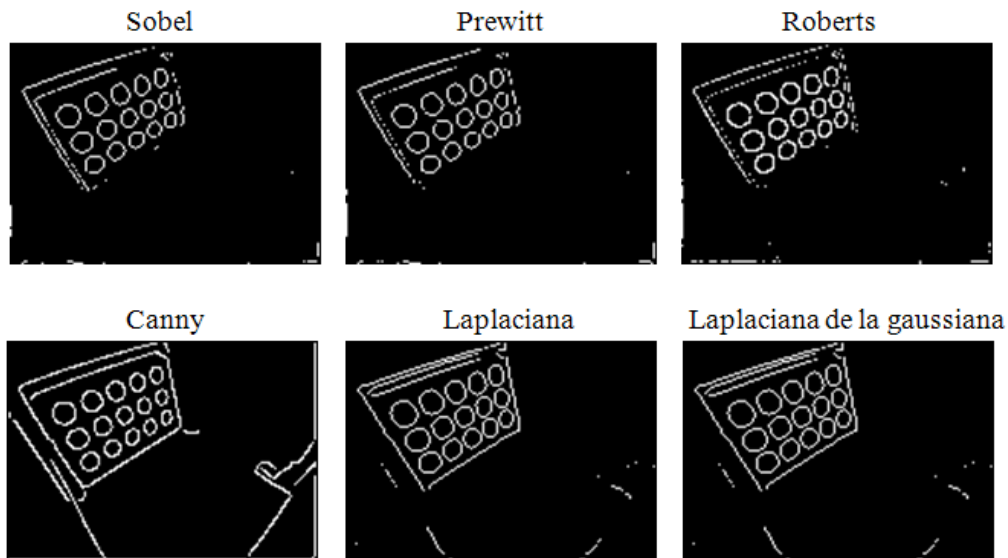


Figura 32. Bordes según los métodos explicados en una imagen cercana.

Se ha observado que el método que proporciona un porcentaje mayor de detección correcta de las elipses del patrón en todos los casos es el método *Canny*, por lo que es el que se va a utilizar. Otros filtros, como el de *Prewitt*, *Robberts* o el *Sobel*, tienden a no cerrar los contornos (Figura 33), y esta característica se ha fijado como necesaria para discriminar entre elipses y otros objetos de la imagen, por lo que su uso se descarta.

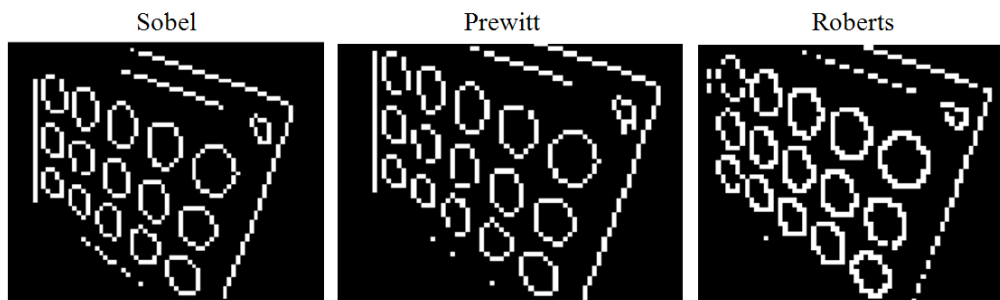


Figura 33. Contornos abiertos en la detección de bordes

Por otro lado, los filtros de Laplaciana y Laplaciana de la Gaussiana sí suelen cerrar los contornos que se detectan, pero presentan otro problema: tienden a hacer que los contornos de las elipses se junten en imágenes en las que el patrón se encuentra alejado y hay menos distancia entre circunferencias, tal como se muestra en los ejemplos de la Figura 34, por lo que estos métodos de segundo orden también van a ser descartados.

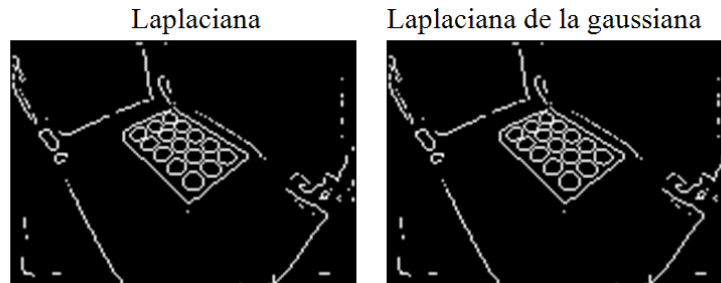


Figura 34. Bordes que juntan los contornos

3.3.3. Selección de formas elípticas

Como se puede apreciar en las figuras de la sección 3.3.2., las imágenes de bordes que se obtienen contienen todos los elementos que aparecen en la imagen. Sin embargo, lo que se busca es quedarse únicamente con las 15 elipses que pertenecen a la proyección de los círculos del patrón. Para ello, en primer lugar hay que establecer unas características que definan estas formas, las diferencien de las demás y fijar unos umbrales entre los que siempre se puedan acotar.

Para ello, primero se etiqueta cada uno de los componentes u objetos conectados de la imagen binaria de bordes mediante la función *'bwlabel'* de MATLAB®, con conectividad 8. A continuación, se obtienen las propiedades necesarias de cada uno de los componentes u objetos de la imagen mediante la función *'regionprops'* aplicada sobre la imagen etiquetada. De todas las características que puede devolver esta función, las que van a obtenerse son:

- Excentricidad. Es el ratio entre el foco de la elipse y la longitud de su eje mayor, y es un valor acotado siempre entre 0 y 1. Como se observa en la Figura 35, si la excentricidad toma un valor de 0 es porque el objeto es totalmente circular, y por el contrario, cuanto más se acerque a 1, mayor va siendo la longitud del eje mayor de dicha elipse.

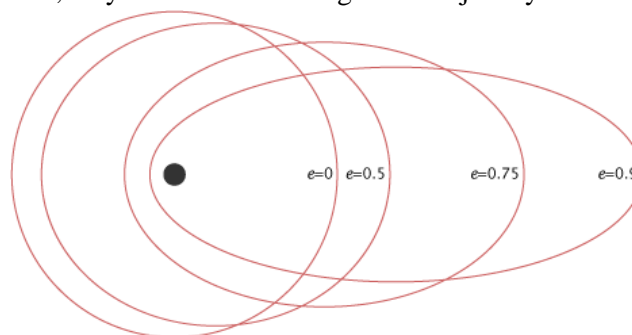


Figura 35. Distintos valores de excentricidad de elipses

- Área. Devuelve el número de píxeles que forman la región. Se va a definir un umbral mínimo y máximo de tamaño que cumplen todas las elipses detectadas en las imágenes independientemente de la posición que tenga el patrón en la escena.

3. Extracción de puntos característicos para la calibración de la cámara ToF

- Número de Euler (E). Esta propiedad indica el número de objetos existentes en la región menos el número de agujeros en dichos objetos. Por tanto, si el objeto está cerrado, como el de la izquierda de la Figura 36, hay solamente un objeto (el contorno) y 1 agujero (el espacio cerrado contenido por este contorno), por lo que el número de Euler valdrá $1 - 1 = 0$. Si el objeto está abierto, como el de la derecha de la Figura 36, hay igualmente un objeto (el contorno), pero no hay ningún agujero, por lo que el número de Euler valdrá $1 - 0 = 1$.

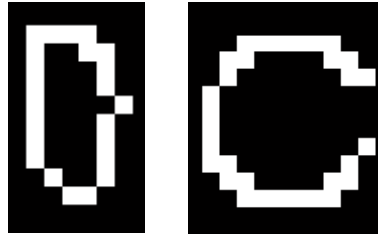


Figura 36. Objeto cerrado $E = 0$ (izquierda) y abierto $E = 1$ (derecha)

- Centroide. Esta característica no va a ser discriminadora para decidir qué objetos son una elipse del patrón, ya que cualquier objeto detectado en la imagen va a tener siempre un centroide, pero sin embargo se va a obtener para partir como primera aproximación para encontrar el centroide de cada elipse en pasos posteriores.

En consecuencia, los valores numéricos que van a restringir los objetos que son candidatos a elipses y los que no van ser:

- Excentricidad: se comprueba que las elipses tienen una excentricidad menor a 0.9 en todos los casos.
- Área: se van a tomar aquellos contornos cuya área sea mayor de 12 píxeles y menor de 95 píxeles, ya que se ha comprobado que proporciona resultados satisfactorios en todos los casos.
- Número de Euler: como las elipses son contornos cerrados y se ha elegido un método para obtener los bordes que tiende a cerrar los contornos, se va a asumir que para que un elemento sea una elipse, el número de Euler ha de ser 0.

De esta manera, se consigue eliminar el resto de contornos de la imagen de bordes, y quedarnos únicamente con aquellos que tienen forma elíptica (Figura 37).



Figura 37. Imagen de bordes (izquierda) e imagen que contiene solo las elipses del patrón (derecha).

Sin embargo, se observa que si hay algún objeto con forma elíptica similar a las del patrón, también es considerada, como aparece en la Figura 38 en la imagen del centro. Para tratar de solucionar este problema se introduce un mecanismo de seguridad en el que se pregunta si se han seleccionado más de 15 contornos de la imagen, y en caso de que se obtenga una respuesta afirmativa, como se sabe que es imposible que se hayan detectado más de 15

círculos si el proceso se ha desarrollado adecuadamente, se llama a una función de eliminación de elipses sobrantes. El concepto en el que se basa este algoritmo es que se asume que las elipses detectadas incorrectamente están más alejadas del resto de elipses de lo que lo están entre sí las elipses que forman parte del patrón. Para ello, el proceso es el siguiente:

- Se calcula la distancia existente entre cada centroide y sus dos centroides más cercanos, contiguos, para todas las elipses detectadas.
- Se calcula la media de todas estas distancias, de manera que se obtiene la distancia media típica que existe entre los centroides del patrón.
- Se eliminan aquellos objetos cuya distancia hasta los dos centroides más cercanos se eleve un número de píxeles determinado (un umbral que se fije) sobre la media.

De esta manera, se consiguen eliminar estos objetos incorrectos, como se muestra en la imagen de la derecha de la Figura 38. Sin embargo, este algoritmo funciona con éxito en este proyecto debido a que por características de la escena durante el proceso de calibración existen pocas formas elípticas detectadas que no pertenezcan al patrón. Si la escena fuera más compleja, habría que diseñar otro tipo de proceso de distinción.

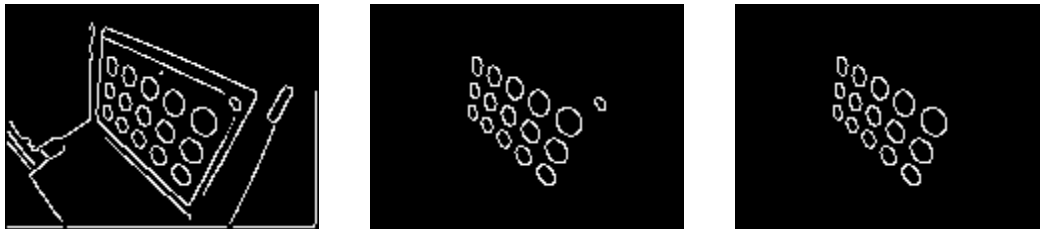


Figura 38. Imagen de bordes inicial (izquierda), elipses detectadas inicialmente (centro) y elipses detectadas tras el algoritmo de eliminación de elipses sobrantes (derecha)

3.3.4. Búsqueda de radio adaptativo

Como se detallará en el apartado 3.3.5, es necesario pasarle como argumento un radio de búsqueda a la función que se encarga de calcular de manera sub-pixélica el centro de las elipses detectadas en torno a la aproximación de centroide hallada. Para ello, inicialmente se trató de buscar un valor fijo de radio para todas las imágenes, y se observó que daba resultados incorrectos si dicha vecindad de búsqueda era mayor o menor de la deseada.

Si el radio es mucho mayor que el radio real de la elipse puede incluirse en la zona de búsqueda parte de alguno de los centroides contiguos provocando que el centro se calcule incorrectamente y se desplace hacia dicho lado. Si por el contrario se toma un radio menor que la elipse, no se está tomando toda la elipse para calcular el centroide, por lo que también se obtiene una medida incorrecta. Se ven ejemplos de esto en la Figura 39.



Figura 39. Imagen sobre la que hallar centros (izquierda), centros hallados con un radio fijo incorrecto (centro) y centros hallados con un radio adaptativo (derecha)

3. Extracción de puntos característicos para la calibración de la cámara ToF

La siguiente medida lógica sería hallar un tamaño para cada imagen, teniendo en cuenta la distancia a la que estuviera el patrón colocado. Sin embargo, este método no da buenos resultados en imágenes en las que el patrón tenga un gran ángulo de inclinación con respecto a la cámara, como ocurre en la Figura 40, porque las elipses más cercanas a la cámara tienen mayor tamaño que las más lejanas, y por tanto, el radio de búsqueda deberá ser distinto para que el resultado sea lo más preciso posible. En consecuencia, si se fija un radio de búsqueda pequeño (imagen central), la búsqueda falla en las elipses grandes, y si se da uno grande (derecha), falla en las elipses pequeñas.

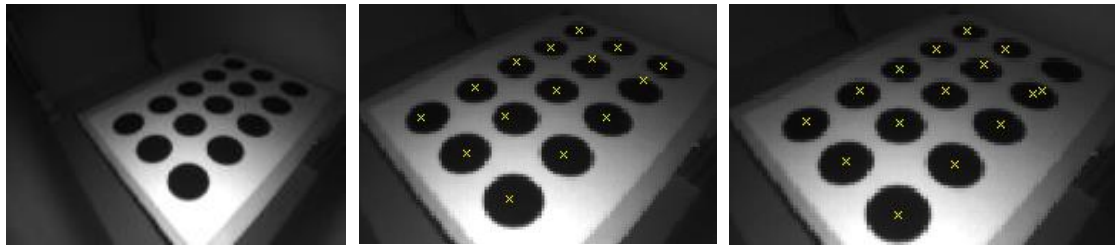


Figura 40. Imagen capturada en la que el patrón presenta una gran inclinación con respecto a la cámara

Por lo tanto, la solución que se decide tomar finalmente es buscar un radio adaptativo para cada elipse de la imagen en función de la distancia existente entre éste y sus centroides contiguos, de la siguiente manera:

- Se toma un centroide y se calcula la distancia existente entre éste y los dos centroides contiguos más cercanos.
- Se halla la media de esas dos distancias calculadas en el paso anterior.
- El radio de búsqueda será la mitad de dicha distancia.
- Se repite el proceso para cada uno de los centroides de la imagen, obteniendo una matriz de radios adaptativos.

De esta manera, se elige siempre el radio óptimo que asegura obtener una vecindad de búsqueda adecuada para el posterior cálculo sub-pixélico del centro, como se muestra en la Figura 41, en la que se calcula de manera correcta los centroides, como no se consiguió hacer con radios fijos en la Figura 40.

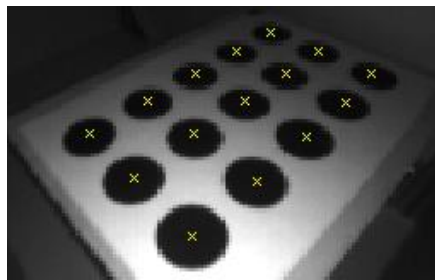


Figura 41. Ejemplo de patrón con cálculo de centroides con radio adaptativo.

3.3.5. Cálculo sub-pixélico del centro de las elipses

Para desarrollar esta tarea se ha utilizado una función sin documentar que proporciona la toolbox de calibración de Yves-Bouguet de MATLAB®, '*centercircfinder.m*'. Esta función realiza un cálculo sub-pixélico del centroide de la circunferencia o elipse a partir de una

aproximación inicial de su posición (obtenida del centroide del objeto que se obtiene mediante la función *regionprops*) y una vecindad de búsqueda que se le proporcione como argumentos.

El proceso que desarrolla, como se anticipó en la sección 3.1, se basa en un método de integración. Básicamente, se asume que pequeñas variaciones producidas en los bordes considerados de la elipse conducen a una mejor estimación del centro de dicha elipse. El proceso que realiza de manera iterativa es el siguiente:

- En primer lugar, se encarga de analizar para la coordenada X e Y del centroide inicial del elemento (o del calculado en la iteración anterior) si la parte decimal se redondea a la unidad superior o inferior, y se crea una máscara con la que se realiza una convolución en la que se modifica ligeramente los niveles de grises de los píxeles pertenecientes a los bordes de las elipses o su zona más cercana en función de la parte decimal de la estimación inicial del centroide. Gráficamente esto resulta en una imagen como la de la derecha de la Figura 42, en la que se observa la modificación producida en los bordes de la imagen con respecto a la imagen inicial, en la izquierda.

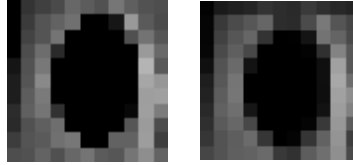


Figura 42. Elipse inicial (izquierda) y elipse tras interpolación (derecha)

- El siguiente paso es calcular los gradientes en la dirección X e Y de las sub-imágenes, de manera que se consiga resaltar los cambios de intensidad que se producen en cada dirección. Como se explicó en el apartado de búsqueda de bordes, las zonas donde se van a producir estos cambios de intensidad es en los bordes de la nueva imagen obtenida.



Figura 43. Gradientes en X e Y

- A continuación, se calcula la posición de la nueva estimación del centroide, de coordenadas $P_c = (x_c, y_c)$, según la ecuación (18), en la que N corresponde a los píxeles de la vecindad tomada en torno a la posición inicial del centroide, y donde $g(u, v)$ son los niveles de gris calculados:

$$\begin{aligned} x_c &= \frac{\sum_{u \in N} \sum_{v \in N} u \cdot g(u, v)}{\sum_{u \in N} \sum_{v \in N} g(u, v)} \\ y_c &= \frac{\sum_{u \in N} \sum_{v \in N} v \cdot g(u, v)}{\sum_{u \in N} \sum_{v \in N} g(u, v)} \end{aligned} \quad (18)$$

- Como último paso de cada iteración del bucle, se comprueba si la diferencia de coordenadas entre la ubicación del centroide inicial y el nuevo calculado P_c es menor que un umbral establecido (la resolución sub-pixélica mínima a la que se pretende ajustar el centroide) o si se ha superado un número establecido de iteraciones máximas para realizar una de estas dos acciones:

3. Extracción de puntos característicos para la calibración de la cámara ToF

- Salir del bucle en caso de que se cumpla alguna de las condiciones anteriores, devolviendo como resultado el valor del último centroide $P_c = (x_c, y_c)$ calculado.
- Volver al paso inicial del bucle, tomando las coordenadas del nuevo centroide $P_c = (x_c, y_c)$ calculado como nueva suposición inicial.

Partiendo de una imagen cuya suposición inicial de los centroides se muestra en la imagen de la izquierda de la Figura 44, en la que a simple vista se puede observar que no corresponden al punto central del círculo, tras ejecutar el algoritmo iterativo explicado se obtienen los centroides de la imagen de la derecha, donde se ha logrado alcanzar una alta precisión en su posicionamiento.

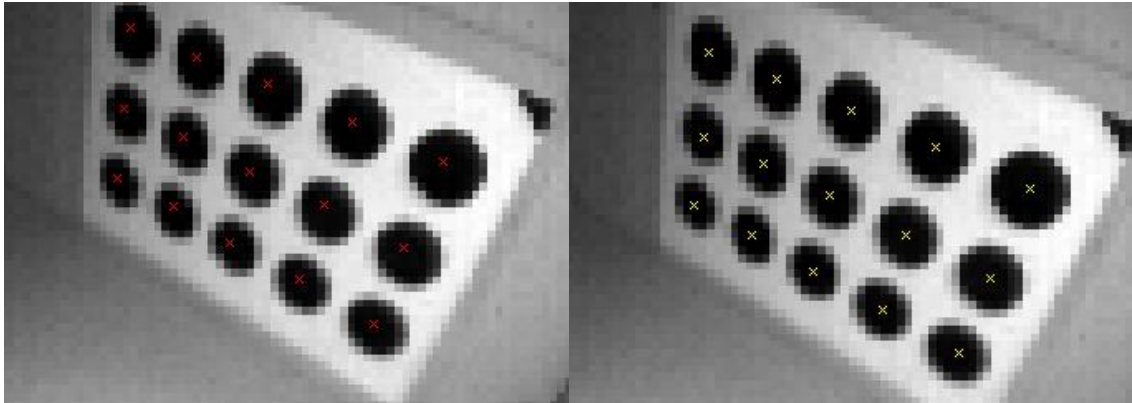


Figura 44. Cálculo de centroides. Centroides iniciales (izquierda) y tras el cálculo sub-pixélico

Sin embargo, es muy importante tener en cuenta que para que este algoritmo funcione adecuadamente ha sido necesario calcular la vecindad de búsqueda óptima mediante el método adaptativo explicado en el apartado 3.3.4, porque si no, lo que ocurre es que en la vecindad de búsqueda se incluye parte de alguno de los centroides adyacentes (Figura 45), y como utiliza los niveles de gris ponderados para calcular el centroide, también tomará en el cálculo los píxeles negros del otro centroide, y hacen que se desplace relativamente hacia ese lado, dando lugar a un cálculo erróneo (cruz marcada en verde).

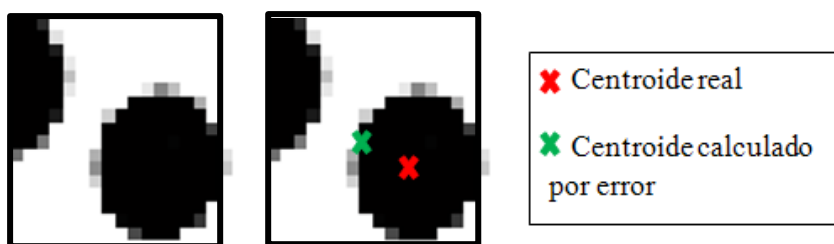


Figura 45. Vecindad tomada para cálculo de centroide (izquierda) y cálculo sub-pixélico (derecha)

Aunque se ha diseñado el programa para que resulte lo más automático posible y la intervención humana sea mínima, puede existir algún caso en el que no se detecte correctamente algún punto de control. Esto puede suceder porque el patrón de calibración se encuentre muy alejado de la cámara en una imagen, y se confundan los contornos de dos elipses distintas como si fuera una sola o porque el patrón esté colocado en un ángulo de rotación elevado con respecto a la cámara (problemas derivados principalmente de la baja resolución de los sensores ToF) y ocurra algo similar. Para estos casos, se trata de posibilitar soluciones rápidas y sencillas al final de la ejecución del algoritmo, las cuales se llevarán a cabo solamente si el usuario considera que

desea mejorar manualmente el resultado de alguno de los centroides obtenidos mediante el proceso automatizado. Estas soluciones son:

- Introducir un nuevo tamaño mínimo o máximo en píxeles de los contornos aceptados como candidatos a elipses pertenecientes al patrón de calibración. Esto puede ser útil si alguna de las elipses tiene un tamaño que se sale de los márgenes establecidos por los parámetros iniciales y no es detectada automáticamente.
- De la misma manera, puede ser que debido a la orientación del patrón, la excentricidad de una elipse sea mayor que la marcada por el límite del valor inicial. También se puede modificar este valor y volver a repetir el proceso.
- Modificar directamente el valor de un píxel concreto *cliqueando* sobre él. Si era píxel de borde, dejaría de serlo, y viceversa. Idealmente no debería darse nunca esta situación, pero en algún caso concreto podría ser necesario:
 - Si el patrón está muy alejado, es posible que el contorno de dos elipses se ‘junten’ en un punto. Se facilita al usuario que pueda solucionar esto simplemente haciendo en *click* en ese punto, eliminándolo de los bordes y consiguiendo dos formas separadas.
 - En alguna ocasión podría ser que el contorno de una elipse no se cerrara, aunque no suele ocurrir porque el método ‘*canny*’ de obtención de bordes funciona bien ante esta característica. Sin embargo, si ocurriese, podría cerrarse el contorno *cliqueando* sobre el píxel que falta, y pasaría a ser un píxel de borde.
- A pesar de que el algoritmo de búsqueda de radio adaptativo es bastante robusto y da buenos resultados, en caso de que el cálculo del centro de la elipse no fuera satisfactorio a partir del radio adaptativo, se da la opción al usuario de seleccionar dicho centroide y poder introducir mediante teclado otro radio de búsqueda.
- Por último, en caso de que el usuario considere que, por cualquier motivo, la imagen no calcula correctamente algún centroide y va a introducir mucho error al proceso de calibración, se posibilita desactivar o rechazar la imagen y descartarla, de manera que la calibración se lleve a cabo sin ella.

3.4. Ordenación de centroides

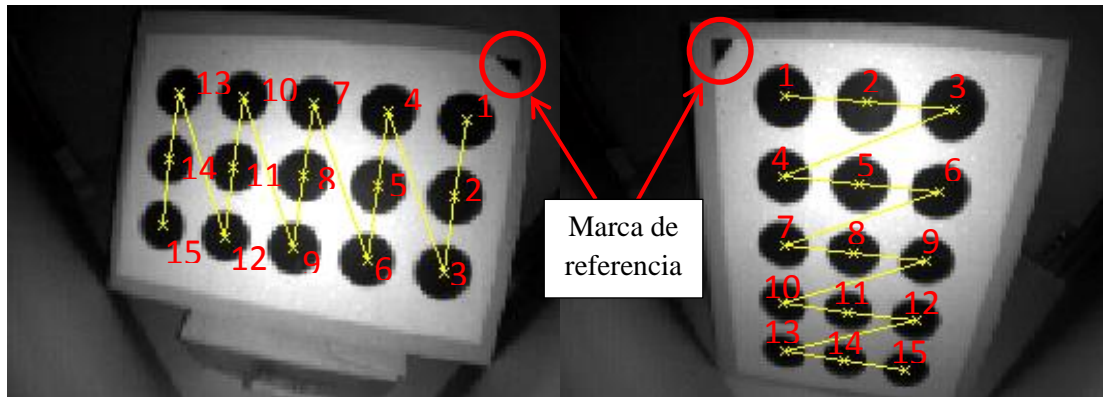
Una vez que todos los centroides han sido hallados correctamente, es necesario que sean ordenados siempre en el mismo orden, independientemente de la posición y orientación que tenga el patrón en la imagen, para que así se pueda establecer correspondencia entre cada centroide hallado y su correspondiente posición en unidades métricas en el plano del patrón, y de esta manera, calibrar. Este paso es importante en el proceso, porque si se toma un orden distinto en alguna imagen, se va a decir que a los centroides calculados les corresponde otra circunferencia de la que en realidad es (con otra posición) y se introduciría un error de calibración al tratar de hacer que los datos incorrectos cuadren con los demás.

Para ello, se ha decidido pedir al usuario que haga click con el ratón sobre la primera elipse del patrón en la imagen calculada, que visualmente se puede reconocer fácilmente por una marca de referencia impresa en la esquina del patrón en la que se encuentra el primer centroide (redondeado en la Figura 46). El orden establecido se muestra también en los ejemplos de la Figura 46 con dos orientaciones distintas del patrón.

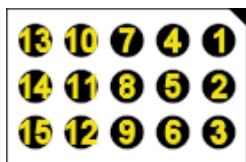
El establecimiento de este orden no es una tarea tan trivial. Básicamente se puede decir que consiste en la búsqueda de las 3 rectas que se pueden trazar uniendo los 5 centroides del

3. Extracción de puntos característicos para la calibración de la cámara ToF

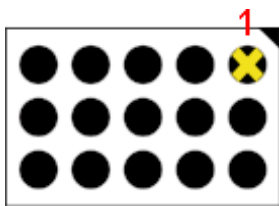
lado largo del patrón, ordenando dichas rectas en función de la distancia al punto inicial, y hallando para cada punto de control cuál es su recta más cercana.



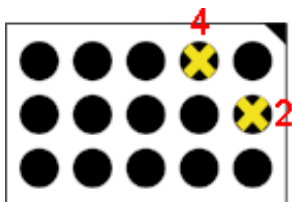
El proceso completo consta de los pasos explicados a continuación:



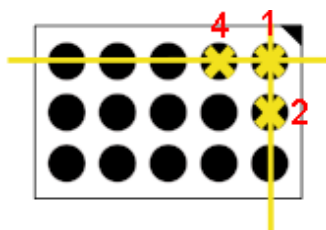
Los centroides inicialmente están ordenados de izquierda a derecha según aparezcan en la imagen, y se busca obtener el orden de centroides mostrados en la Figura 47.



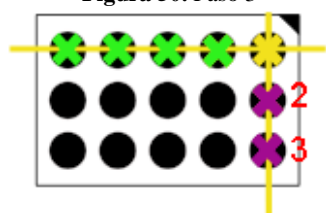
En primer lugar, se detecta cual es el centroide número 1, que será el que se sitúe más próximo a la marca existente en la esquina de este patrón y colocada para este fin.



A continuación, se calcula la distancia al resto de centroides y se toman los dos más cercanos, que idealmente deberían ser los centroides 2 y 4.



Se traza una recta que pase por el centroide 1 y 4, y otra recta entre los centroides 1 y 2. El objetivo es detectar cuál es el lado largo y cuál el corto del patrón.



Se calcula la distancia de cada centroide a las dos rectas, y los centroides que presenten una distancia menor a un determinado umbral de píxeles, se consideran pertenecientes a las rectas. De esta manera, podemos detectar cuál es el lado 'largo' del patrón (con 5 centroides en la recta) y el 'corto' (con 3). Tomamos, por tanto, el lado 'corto' del patrón, y así detectamos los centroides 2 y 3.

Figura 51. Paso 4

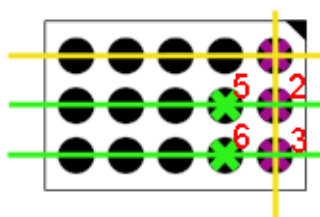


Figura 52. Paso 5

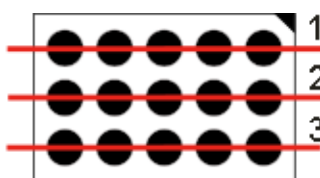


Figura 53. Paso 6

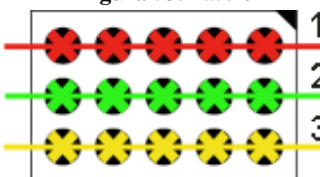


Figura 54. Paso 7

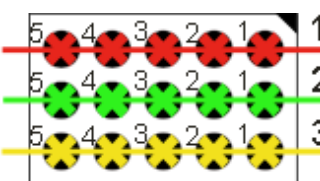


Figura 55. Paso 8

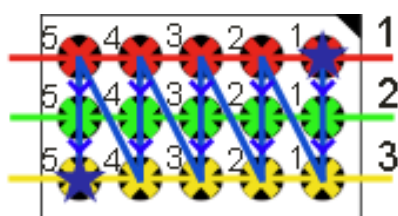


Figura 56. Paso final

El siguiente paso es trazar las 3 rectas que pasan por los centroides del lado ‘largo’ a partir de haber hallado los centroides 2 y 3. Para ello, se busca el centroide más cercano al 2, que es el 5, y se traza una recta entre ambos. A continuación, se busca el centroide más cercano al 3, que es el 6, y se traza recta entre ambos.

En el paso anterior, se obtuvieron las 3 rectas. Ahora, se las numera en orden ascendente del 1 al 3 en función de la distancia de cada recta al centroide inicial, el número 1.

Ahora, se calcula la distancia de los centroides a las rectas creadas, de manera que se asocien cada uno de los centroides, de izquierda a derecha (el orden inicial que tenían), a una de las rectas. Por tanto, idealmente ya están todos los centroides contenidos en 3 arrays de 5 elementos.

Como mecanismo de seguridad, se ordenan los 5 centroides de cada array en función de la distancia al centroide inicial de su recta (1, 2 ó 3).

Por último, se ordenan finalmente los centroides a través de los arrays creados, en orden ascendente de elementos en primer lugar, y a continuación de filas, según se muestra en la Figura 56, comenzando por el centroide 1 de la fila 1, y siguiendo las flechas hasta acabar con el centroide 5 de la fila 3.

Con este método se logra tener siempre los centroides ordenados independientemente de la posición y la orientación que presente el patrón con respecto a la cámara en la imagen capturada, obteniendo los resultados que se mostraban en la Figura 46.

Sin embargo, hay un aspecto importante a tener en cuenta para que siempre funcione adecuadamente: en el paso 2, cuando se trata de buscar los dos centroides más cercanos al centroide número 1, idealmente el resultado que se busca es obtener los centroides 2 y 4. Sin embargo, dependiendo de la perspectiva de la imagen y debido a que hay zonas con alta distorsión, es posible que en alguna imagen el centroide 5 esté más cerca del 1 que el 2 o el 4 (Figura 57), y se tome este por error, desencadenando en una ordenación totalmente incorrecta de los puntos.

3. Extracción de puntos característicos para la calibración de la cámara ToF

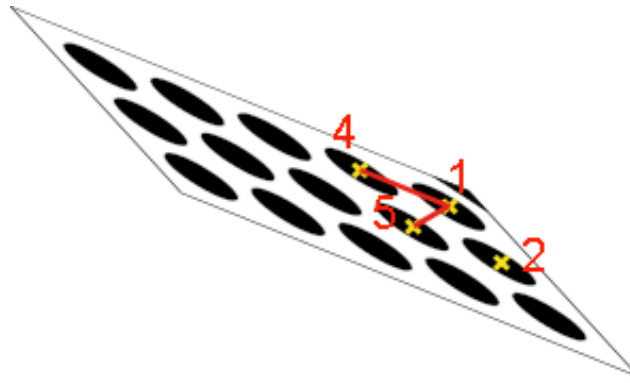


Figura 57. Ejemplo de colocación del patrón en el que el centroide 5 está más cerca que el 4

Para estos casos, hay un mecanismo de seguridad previsto que lo evita. Los pasos de los que consta son:

- En primer lugar, calcula la distancia de todos los centroides al centroide 1, y toma los 3 centroides más cercanos, que han de ser el 2, 4 y 5.
- De estos 3 centroides, toma el más cercano al 1, que será en todos los casos el número 4 o el 2, que va a ser llamado ‘centroide base’ (supongamos que en este caso es el centroide 2 de la Figura 57).
- Ahora, hay que discriminar entre el 2º y el 3º centroide más cercanos para reconocer cuál de ambos hay que elegir (siguiendo con la suposición, centroides 4 y 5 de la Figura 57). Esto solo va a ser necesario cuando las diferencias de distancias de cada uno de ellos con el centroide 1º sea menor que un cierto umbral, que es cuando se pueden dar este tipo de errores. Es decir, si se establece el umbral en 1.5 píxeles, y tenemos que la distancia entre los centroides 1-5 es de 5 píxeles y la 1-4 es de 4, como se está dentro del umbral, es necesario realizar este paso.
- Por lo tanto, si la diferencia es menor que dicho umbral, se va a calcular la distancia entre cada uno de estos 2 centroides y el centroide base, y el que esté más lejos del centroide base (el 2 para el ejemplo que estamos siguiendo), será el que está más cerca del centroide número 1. Aplicado a la suposición sobre la Figura 57, el centroide que está más lejos del base (2), es el 4, que efectivamente es el que debía ser elegido.

Un mecanismo de seguridad basado en la misma técnica se ha de llevar a cabo durante el paso 5 para que la perspectiva o la distorsión de la imagen no provoquen que se pueda llegar a escoger el centroide 6 para formar la recta con el 2, o el centroide 5 para el 3. Además, debido a que se pretenden cubrir todas las posibles circunstancias, aunque se prevé que los centroides siempre se ordenen correctamente, se ha programado una alternativa en la que el usuario pueda indicar que la ordenación es incorrecta en caso de que así ocurra, y poder modificarla.

En la Figura 58 se muestran ejemplos del correcto funcionamiento del método de ordenación de centroides en varias imágenes tomadas por las cámaras.



Figura 58. Centroides ordenados en varias imágenes

4. Calibración de las cámaras

La calibración de una cámara es la tarea de calcular los parámetros que permiten expresar matemáticamente el modelo que transforma un punto 3D de una escena del mundo real a su punto correspondiente en el plano imagen de la cámara. En general, se podrían clasificar los métodos de calibración existentes en dos categorías principales:

- Calibración fotogramétrica: se lleva a cabo tomando imágenes de un objeto de calibración bidimensional o tridimensional cuyas medidas son perfectamente conocidas. Es en la categoría que se enmarca el proceso de calibración de este trabajo.
- Auto-calibración (self-calibration): se refiere al proceso de cálculo de los parámetros de calibración utilizando únicamente información de las imágenes capturadas por la cámara. La propia geometría de la escena genera restricciones de cómo los puntos capturados deberían aparecer en la imagen desde distintas perspectivas. Por lo tanto, moviendo la cámara en una escena estática, la rigidez de la escena genera dos nuevas restricciones por cada imagen capturada. Tomando imágenes con una misma cámara desde distintas posiciones es posible calcular los parámetros internos y externos de ésta y realizar una reconstrucción 3D [15].

Los métodos de auto-calibración tienen la ventaja de que no es necesario el uso de un patrón de calibración. Sin embargo, presentan otra dificultad, y es que encontrar correspondencias de manera automática en varias imágenes no es una tarea sencilla. Por tanto, se va a elegir un método perteneciente a la primera clase.

La calibración de las cámaras se va a realizar a través de las funciones programadas en la toolbox de MATLAB® de Yves-Bouquet [12], que está basada en el método Zhang [4] [5]. Esta toolbox fue elegida con respecto a otras disponibles públicamente online (toolbox de Zhang's [13], Heikkilä's [14]) por ser la más comúnmente utilizada para cualquier trabajo que requiera un proceso inicial de calibración y por proporcionar buenos resultados. Sin embargo, esta toolbox no puede ser utilizada directamente ya que muchas de sus funciones no proporcionan buenos resultados para las cámaras con las que se está trabajando, simplemente se sacará partido de algunas de sus funciones que nos pueden ser útiles.

4.1. Método Zhang

El método Zhang trata de calibrar una cámara utilizando un patrón de dos dimensiones que tenga unos puntos característicos conocidos (un patrón como el de la Figura 17). Esta técnica de calibración es muy flexible ya que permite calibrar una cámara tomando imágenes del patrón situado en varias posiciones y distancias (al menos 2) relativas al sensor, sin necesidad de tener que conocer ni la distancia a la que está ni su orientación con respecto a la cámara. Simplemente es necesario conocer con precisión la posición relativa de dos puntos de control entre sí en unidades métricas del mundo real.

Este método fue elegido por los buenos resultados que presenta en diferentes estudios realizados previamente y por la facilidad de programación que presenta. Además, se pretende también en este trabajo testar su funcionamiento con cámaras de menor resolución y calidad para calibración de cámaras de tiempo de vuelo de bajo coste.

El primer paso de esta técnica consiste en resolver el modelo de la cámara pinhole explicado en la sección 2.3, cuya ecuación se expresa a continuación, sabiendo que K es la matriz intrínseca de la cámara, r_1, r_2 y r_3 las columnas de la matriz de rotación y t la matriz de traslación. El método Zhang también modela la distorsión radial.

De aquí en adelante, nos referiremos a un punto en 2D como $p = [u, v]^T$, y a un punto en 3D como $P = [X, Y, Z]^T$. Usaremos el símbolo \sim para indicar los vectores aumentados añadiendo un 1 como último elemento: $\tilde{p} = [u, v, 1]^T$ y $\tilde{P} = [X, Y, Z, 1]^T$. El parámetro s es un factor de escala y R y t los parámetros extrínsecos. Se modela la cámara según el modelo pinhole teniendo en cuenta estos parámetros según la ecuación (19):

$$s\tilde{m} = K \cdot [R \quad t] \cdot \tilde{P} \quad (19)$$

$$s \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & \gamma & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{pmatrix} \cdot \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (20)$$

Para más información de cómo la el método Zhang resuelve y calcula los parámetros de la calibración, referirse al ANEXO II: Método Zhang.

4.2. Toolbox Yves-Bouquet MATLAB®

Cuando se inicializa la toolbox de MATLAB® mediante la ejecución de la sentencia '*calib_gui*' en la barra de comandos, se abre una figura que muestra el menú de calibración de la Figura 59.



Figura 59. Interfaz gráfica de la toolbox Yves-Bouquet

Mediante esta herramienta se pueden calibrar cámaras a través de imágenes RGB o en escala de grises capturadas por ellas, en las que aparezca un patrón de calibración de tipo damero situado en varias posiciones de la escena. Los puntos de control que se buscan mediante esta toolbox son las esquinas (mediante 'Extract grid corners'). Sin embargo, como se razonó en la sección 3.2, en este trabajo se ha decidido finalmente utilizar un patrón con formas circulares en el que se buscarán centroides, por lo que la función que no se va a utilizar la función de búsqueda de esquinas y principalmente se va a aprovechar la función '*go_calib_optim_iter.m*', que es la que se encarga de aplicar el método Zhang para hallar los parámetros de calibración intrínsecos y extrínsecos de la cámara a partir de los centroides calculados en la imagen y de las correspondientes coordenadas 3D conocidas del grid del patrón.

Esta función comienza hallando los valores intrínsecos y extrínsecos iniciales sin iterar, y a continuación realiza un proceso iterativo de optimización con la técnica de descenso por el gradiente, para finalmente proporcionar unos resultados más precisos y refinados de los parámetros de calibración.

También han sido útiles otras funciones de esta toolbox como la de leer y tomar las imágenes disponibles en el directorio para la calibración y las de guardar o cargar datos de calibración, así como la función de generar las nuevas imágenes con la distorsión corregida. También podría ser útil, si el usuario lo desea, utilizar la función de visualización de error en la re proyección de los puntos característicos (Figura 60), por si se quiere eliminar alguna imagen que presente mucho error o corregir alguna imagen en cuestión (mediante ‘*Analyse error*’), de manera que la calidad de la calibración aumente o visualizar las posiciones de los patrones de cada imagen con respecto al sistema de coordenadas de la cámara (mediante ‘*Show Extrinsic*’).

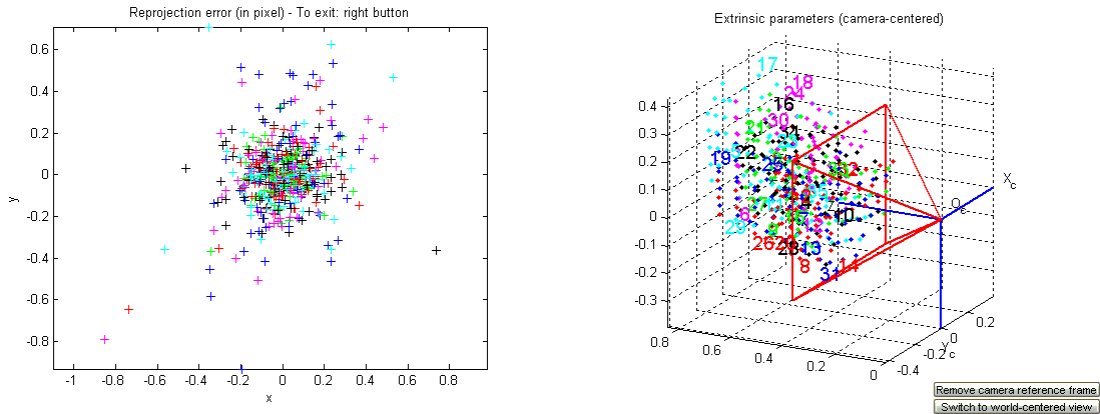


Figura 60. Error de reproyección en píxeles (izquierda) y patrones en 3D (derecha)

4.3. Comparación con distintos patrones

Una vez que ha sido explicado el proceso de calibración, van a mostrarse los resultados obtenidos con los diferentes patrones de calibración. Se va a calcular una medida para comprobar la linealidad de los puntos de calibración, como se explica en el apartado 4.3.1. *Medida de la linealidad*, y a continuación pasarán a compararse directamente los resultados en el apartado a través de este método en la sección 4.3.2. *Comparación de resultados con distintos patrones mediante linealidad*.

4.3.1. Medida de la linealidad

Para comprobar de manera sencilla la precisión de la calibración se puede utilizar un indicador de linealidad, que es útil ya que los puntos de control del patrón idealmente deberían estar alineados entre sí horizontalmente en 3 filas de 5 centroides y verticalmente en 5 columnas de 3 centroides, independientemente de la posición que ocupe el patrón en la imagen capturada. Sin embargo, los efectos de la distorsión provocan que no ocurra de esta manera. De hecho, las cámaras ToF con las que se está trabajando presentan una distorsión bastante alta, la cual se muestra de manera más pronunciada sobre todo en las zonas cercanas a las esquinas y bordes de la imagen, y ni siquiera responde a un único modelo barril o cojín (Figura 12). Esto provoca que los puntos que, idealmente, deberían formar una recta perfecta en las imágenes, no lo hagan, y estas rectas se curven o se deformen.

Sin embargo, tras calibrar las cámaras y hallar los parámetros de distancia focal, centro óptico y distorsión, se puede corregir la distorsión presente en las imágenes (según la ecuación (9)), y el resultado se acercará más a la imagen ideal que se esperaba. Por tanto, si se caracteriza la linealidad de los puntos de control antes y después de corregir la distorsión de las imágenes,

se puede cuantificar el aumento de la linealidad mediante la calibración, y de esta manera comprobar si ha sido efectiva.

El primer paso es calcular las coordenadas que tendrían los centroides sin distorsión a partir de los puntos de control hallados en las imágenes obtenidas. Para conseguirlo, se toma la matriz ampliada con las coordenadas ampliadas de dichos píxeles en dos dimensiones $\tilde{p} = (u, v, 1)^T$ y las multiplicamos por la inversa de la matriz intrínseca K :

$$p = K \cdot P \rightarrow P = K^{-1} \cdot p \rightarrow \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}^{-1} \cdot \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \quad (21)$$

De esta manera se realiza la conversión de coordenadas en píxeles a unidades métricas. El siguiente paso es aplicar las expresiones de (9). Para ello, se utiliza la función '*apply_distortion.m*' que proporciona la toolbox de MATLAB®. Esta función tiene una particularidad que ha de ser comentada: se observa que si se le pasa como argumento los coeficientes de distorsión radiales k_1 , k_2 y k_3 y tangenciales p_1 y p_2 directamente obtenidos durante la calibración, en lugar de corregirse la distorsión, justamente ocurre lo contrario, aumenta. Resulta que la función de calibración devuelve estos coeficientes con el signo contrario, y si solamente se utiliza la toolbox de MATLAB®, en algún punto de las funciones se compensa el signo, y finalmente entrega el resultado correcto. Esto es un aspecto importante a tener en cuenta en la toolbox, y es que se supone que el resultado final de cualquier proceso realizado únicamente con sus funciones proporciona un resultado esperado, mientras que la utilización parcial de partes de código puede derivar en errores inesperados como éste si no se analiza correctamente qué hace el código y qué parámetros y en qué formato deben proporcionarse. En consecuencia, para que se corrija la distorsión correctamente, se cambia el signo del vector de coeficientes de distorsión:

$$k_c = (k_1 \quad k_2 \quad p_1 \quad p_2 \quad k_3) \rightarrow k'_c = -k_c \quad (22)$$

Por lo tanto, se le pasa a la función '*apply_distortion.m*' los centroides $P = (X, Y)^T$ y los coeficientes de distorsión k'_c , y la función devuelve los puntos sin distorsión P_u en unidades métricas. Para obtener los puntos de control sin distorsión p_u de nuevo en coordenadas pixélicas, se proyectan al plano imagen según la ecuación (23):

$$p_u = K \cdot P_u \rightarrow \begin{pmatrix} u_u \\ v_u \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} X_u \\ Y_u \\ 1 \end{pmatrix} \quad (23)$$

Además, no solamente se van calcular los centroides sin distorsión sino que también se aplica esta misma teoría para calcular la imagen final con la distorsión corregida en todas sus zonas. En la Figura 61 se muestran algunos ejemplos de imágenes reales e imágenes con la distorsión eliminada. Como se puede apreciar, en las imágenes reales hay elementos en la escena que deberían ser una línea recta, y aparecen curvos o deformados (como los bordes del patrón, las esquinas de la caja de la escena...), y en las imágenes corregidas (abajo), estos elementos pasan a ser líneas rectas.

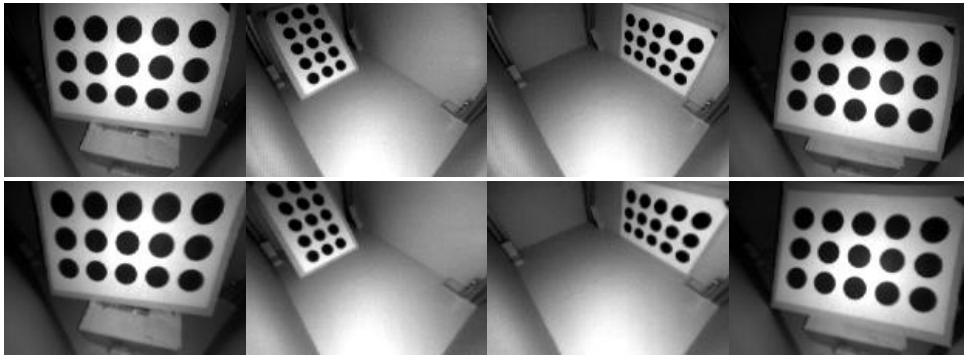


Figura 61. Ejemplo de pares de imágenes reales con distorsión (arriba) y su correspondiente imagen con la distorsión corregida (abajo)

En la Figura 62 se muestra un ejemplo representando también los centroides calculados. Arriba a la derecha está la imagen real capturada con la cámara, que presenta distorsión, en el centro se muestra la imagen tras realizar la corrección de la distorsión, y a la izquierda los centroides con y sin distorsión proyectados sobre la imagen inicial. Abajo, se hace un zoom en la zona del patrón que presenta la información de los dos grupos de puntos característicos y se representan en verde los centroides encontrados en la imagen inicial tomada por la cámara, y en amarillo los corregidos con la distorsión eliminada.

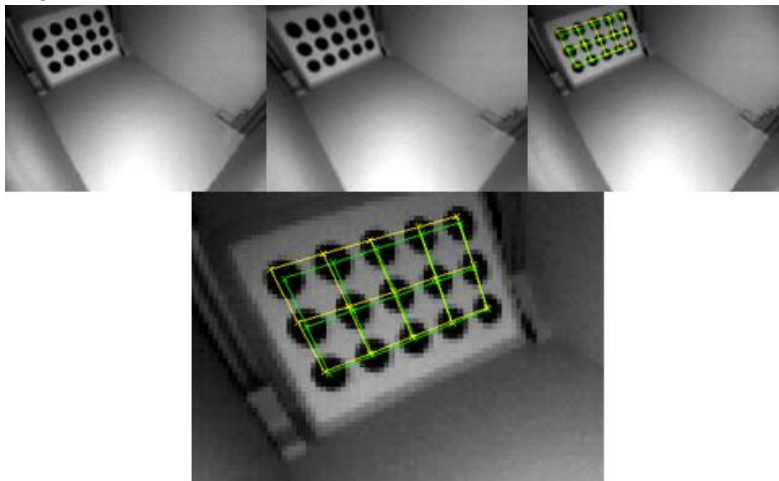


Figura 62. Centroides con y sin distorsión representados sobre una imagen capturada

Los centroides que aparecen dibujados en las imágenes de la Figura 62 son los que se van a utilizar para determinar la efectividad de la calibración como un primer paso a través de su linealidad. Más adelante, con ayuda de la información de profundidad de las ToF, y después con la calibración estéreo y la reconstrucción 3D se podrán realizar más medidas de conformidad con la calibración elegida, comprobando si las distancias obtenidas son las que realmente hay en el mundo real.

Para calcular la medida de linealidad solamente son necesarios los centroides iniciales y los que han sido corregidos. En primer lugar, se toman los centroides iniciales y se hallan las rectas ficticias que mejor se ajusten a ellos y que los unan tanto en la dirección horizontal (3 rectas de 5 centroides), como en la dirección vertical del patrón (5 rectas de 3 centroides), como se muestra en la Figura 63. Idealmente, si los centroides han sido perfectamente calculados en la imagen y la lente no introdujera ningún tipo de distorsión, estas rectas ficticias deberían pasar exactamente por todos los centroides asociados a ellas. Sin embargo, debido al efecto de la

distorsión, estas rectas ficticias se curvan, y la distancia que existe desde cada punto a su recta ideal correspondiente aumentará. Esta distancia se va a almacenar para cada centroide.

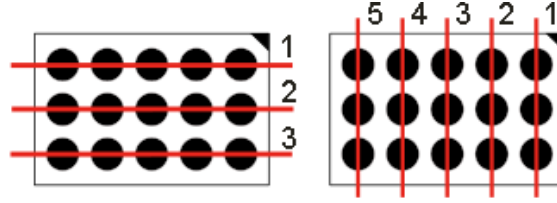


Figura 63. Cálculo de linealidad. Rectas horizontales (izquierda) y verticales (derecha)

A continuación, se realiza el mismo proceso para los centroides cuya posición ha sido corregida al eliminar la distorsión, hallando la distancia de cada uno de ellos a sus rectas. Si se ha corregido la distorsión correctamente, en este caso los centroides deberían estar más cerca de las rectas ficticias calculadas que en la imagen inicial, por lo que las distancias van a ser menores que en el caso anterior, y esto indicaría que la linealidad ha mejorado, y es un indicio de que se han calculado con éxito los parámetros de distorsión mediante la calibración.

Por último, se halla el ratio entre estas distancias, las halladas con los centroides iniciales de la imagen capturada ($dist_{\text{distorted}}$) y con los corregidos ($dist_{\text{undistorted}}$), tal y como se muestra en la ecuación (24):

$$\text{linealidad} = \frac{dist_{\text{distorted}}}{dist_{\text{undistorted}}} \quad (24)$$

Si el resultado es correcto, ocurriría lo siguiente:

$$dist_{\text{distorted}} > dist_{\text{undistorted}} \rightarrow \text{linealidad} > 1 \quad (25)$$

Por lo tanto, si el proceso de calibración ha sido llevado a cabo satisfactoriamente y los parámetros de distorsión que modelan los errores que introduce la lente han sido calculados correctamente, el coeficiente de linealidad medio calculado será mayor que la unidad. Además, cuanto mayor sea, más corrección de linealidad se habrá producido. Para poder valorar lo que este parámetro aporta realmente, han de tenerse en cuenta ciertos aspectos:

- Es importante que el proceso de búsqueda de centroides sea muy preciso (por debajo del píxel), porque si se parte de una posición inicial de los centroides que se aleje de la real, se está realizando el proceso partiendo con datos erróneos y ninguna de las medidas que se puedan calcular tendría sentido.
- Se va a medir la linealidad tanto en la dirección vertical y como en la horizontal del patrón, es decir, en el lado ‘largo’ y en el ‘corto’. En general, el coeficiente de medida horizontal va a ser más significativo que el vertical, ya que incorpora información de una recta que se trata de ajustar a 5 centroides, y en el segundo, solamente a 3 centroides, por lo que el ajuste será más sencillo, y la medida de linealidad que se obtendrá será menos significativa que si el número de puntos fuera mayor.
- Debido a que la distorsión es bastante mayor en las zonas de los bordes y las esquinas de las imágenes, el hecho de que la medida de linealidad sea la unidad no significa que la calibración no haya mejorado las imágenes. Si en una imagen el patrón está situado en la zona central, estará poco distorsionado y probablemente la medida de linealidad sea cercana a la unidad, pudiendo incluso ser ligeramente menor a la unidad, pero estos casos no serán significativos para concluir en una mejora o un empeoramiento de la imagen.

- La linealidad se va a hallar en cada cámara con todo el set de imágenes disponibles, que será en torno a 35 imágenes en cada caso. El hecho de que en una imagen el coeficiente de linealidad sea menor que la unidad no indica necesariamente que la calibración haya sido mal realizada. Además del caso expuesto en la conclusión previa, si la mayoría de las imágenes proporcionan un coeficiente mayor que la unidad, y algún caso puntual presenta un coeficiente bastante menor que la unidad, podría simplemente significar que los centroides iniciales en esa imagen fueron mal hallados en la etapa de búsqueda de puntos de control en las imágenes.

4.3.2. Comparación de resultados con distintos patrones mediante linealidad

Inicialmente, el fabricante proporciona al usuario unos parámetros de calibración (focal, centro óptico y distorsión) en archivo de datos en una memoria portable con la compra de la cámara, y asegura que son suficientemente precisos para trabajar con las cámaras. Sin embargo, no es posible que distribuyan una cámara perfectamente calibrada con una lente que se puede mover para enfocar la escena. En realidad lo que facilita el fabricante son unos datos aproximados, que si no se pretende realizar reconstrucción 3D precisa ni fundir datos de varias cámaras, podría ser suficiente para el desarrollo de una aplicación sencilla. En cambio, en el trabajo desarrollado es necesario alcanzar la mayor precisión posible en los datos de calibración para poder reconstruir con mucha exactitud, por lo que es totalmente necesario calcular unos datos de calibración más precisos que los del fabricante.

Aparte de los datos del fabricante, también se van a comparar los de dos dameros y dos patrones con circunferencias, analizando de esta manera la eficiencia en la detección al utilizar motivos impresos de distintos tamaños. Además, no solamente se van a mostrar los datos de los dameros obtenidos mediante MATLAB®, sino que también se va a mostrar algún ejemplo de los que proporciona OpenCV con los dameros. En la Figura 65 se muestra una imagen de los patrones comparados. Se quiso comparar también una calibración realizada con OpenCV en la que se detectasen los centroides de las circunferencias, pero debido a que no se detectaban estos centroides en la mayoría de las imágenes, no fue posible. A continuación se muestra una tabla con los resultados numéricos de las calibraciones que se van a utilizar, y más adelante se verán las comparaciones entre cada uno de ellos que permitirán discernir cuál produce mejores resultados de entre todos ellos.

CALIBRATION PARAMETERS	fx	fy	cx	cy	k1	k2	p1	p2	k3
1 - Fabricante	104.119	103.588	81.949	59.43	-0.222609	0.063022	0.002865	-0.001446	0
2 – Patrón damero pequeño - OpenCV	88.310	88.695	76.840	53.697	-0.15618	0.20894	-0.00240	-0.00257	0
3 - Patrón damero grande - MATLAB	90.718	90.315	71.631	60.073	-0.10857	-0.05751	0.00510	-0.01853	0
4 - Patrón damero grande - OpenCV	90.72	90.32	71.61	60.07	-0.1082	-0.05757	0.00513	-0.001856	0
5 - Patrón de círculos pequeños	89.714	89.266	76.837	55.725	-0.08486	-0.01007	-0.00247	-0.00281	0
6 - Patrón de círculos grandes	87.385	87.426	77.214	55.8363	-0.06447	-0.04629	-0.00038	0.00186	0

Figura 64. Resultados de los parámetros de algunos ejemplos de calibración con distintos métodos y patrones

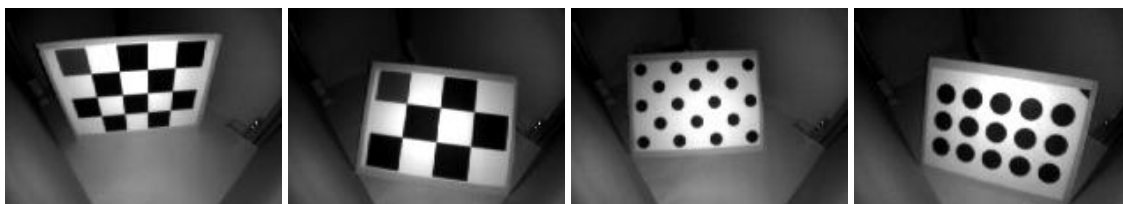


Figura 65. Patrones utilizados para la calibración. De izquierda a derecha: damero grande, damero pequeño, círculos pequeños y círculos grandes.

Inicialmente, en la parte superior de la Figura 66 se muestra una imagen en la que se simplemente mediante inspección visual se pueden apreciar los efectos de la distorsión, y debajo de esta imagen se muestran las correcciones de distorsión realizadas por cada uno de los parámetros obtenidos por los procesos de calibración expuestos en la Figura 64. En todos los casos, como se muestra, se va a realizar la corrección sobre imagen mostrada en la parte superior, que es una imagen en la que se captura el patrón que finalmente será elegido (patrón de círculos grandes) como más eficiente para la calibración de este tipo de cámaras de baja resolución.

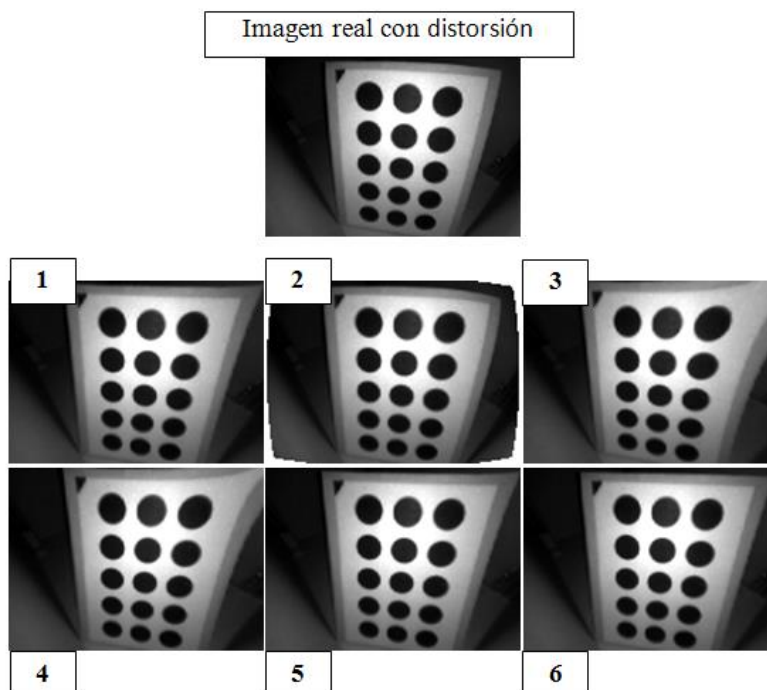


Figura 66. Distintas correcciones de distorsión los parámetros hallados por cada método (1-6) de izquierda a derecha y de arriba a abajo

Como se observa visualmente en la Figura 66, el método del fabricante corrige la distorsión de manera aceptable. Sin embargo, los parámetros intrínsecos que proporciona no son los reales, ya que la posición de la lente varía con respecto a la que había cuando el fabricante realizó la calibración de las cámaras, por lo que las coordenadas tridimensionales que se obtendrían a través de reproyectar con la matriz intrínseca y las distancias de la medida de la cámara ToF podrían no ser las correctas. Por lo tanto, esta calibración no va a ser considerada inicialmente como válida. Los 3 métodos de calibración mediante dameros (2, 3 y 4), tanto en MATLAB® como en openCV, visualmente proporcionan imágenes que presentan una distorsión mayor a la de la imagen capturada por la cámara, por los que simplemente por este

hecho se podría afirmar que no han calculado apropiadamente los parámetros durante la etapa de calibración.

Los métodos 5 y 6, en los que el patrón utilizado tiene impresos motivos circulares sobre él y el método de cálculo de puntos de control es el explicado en la sección 3 del presente trabajo, son los que visualmente proporcionan mejores resultados, ya que se observa como las zonas de la imagen que deberían ser rectas en la imagen real capturada y no lo eran por efecto de la distorsión, se transforman en lo que a priori parecen rectas (visualmente).

Tras una aproximación intuitiva al problema, se van a mostrar resultados empíricos del cálculo de la linealidad, para efectivamente poder demostrar que el método 6 es el que proporciona mejores resultados de linealidad. Para ello, se captura un set de 35 imágenes con una de las cámaras y se calculan sus correspondientes centroides con precisión sub-pixélica, posteriormente se elimina la distorsión de dichos centroides con los parámetros de distorsión calculados por cada uno de los 6 métodos de calibración explicados y se realiza el proceso que se muestra a continuación:

- En primer lugar, se halla la distancia existente entre cada centroide y su recta ficticia correspondiente, como se explicó en el apartado 4.3.2. Se realiza esto para todos los centroides encontrados en la imagen capturada, la que presenta los efectos de la distorsión, y se halla la media aritmética de todas las distancias, tanto a las rectas verticales como a las horizontales:

$$\begin{aligned} d_{\text{dist}_H} &= \sum_{n=1}^{15} d_{\text{centroide distorted}_H} \\ d_{\text{dist}_V} &= \sum_{n=1}^{15} d_{\text{centroide distorted}_V} \end{aligned} \quad (26)$$

- Se halla a continuación la distancia entre cada centroide corregido y su correspondiente recta ficticia para los centroides hallados tras la corrección de distorsión, así como su media aritmética:

$$\begin{aligned} d_{\text{undist}_H} &= \sum_{n=1}^{15} d_{\text{centroide undistorted}_H} \\ d_{\text{undist}_V} &= \sum_{n=1}^{15} d_{\text{centroide undistorted}_V} \end{aligned} \quad (27)$$

- Se calcula el coeficiente de linealidad que se explicó en la ecuación (24) a partir de estas dos distancias calculadas en las ecuaciones (26) y (27):

$$\begin{aligned} \text{lin}_H &= \frac{\text{dist}_{\text{dist}_H}}{\text{dist}_{\text{undist}_H}} \\ \text{lin}_V &= \frac{\text{dist}_{\text{dist}_V}}{\text{dist}_{\text{undist}_V}} \end{aligned} \quad (28)$$

Se realiza este proceso repetitivamente para cada una de las 35 imágenes en cada uno de los 6 métodos, y se obtiene la tabla que se muestra en la Figura 67, en la que cada fila corresponde a una de las 35 imágenes, y en las distintas columnas se muestran los 6 métodos, y para cada uno de ellos, los coeficientes de linealidad horizontal (lin_H) y vertical (lin_V) obtenidos en cada imagen. Las conclusiones principales que se obtienen de estos datos son:

- En primer lugar, principalmente hay que centrarse en el indicador horizontal del patrón lin_H como elemento discriminatorio, ya que contiene información de 5 centroides, como ya se explicó, frente a lin_V que solo contiene de 3, y proporcionará información más significativa.

- En segundo lugar, en la parte de debajo de la tabla aparece el número de imágenes en las que se ha aumentado la linealidad. Teniendo en cuenta que las imágenes tomadas son comunes en todos los métodos, el que mejore más imágenes se podrá afirmar que funciona mejor en cualquier posición de la escena a capturar, y por tanto es lo que se busca. El método 6 es el que mejor funciona, con 33 imágenes mejoradas de 35, y las dos imágenes que no son mejoradas, se observa que tampoco lo son en la mayoría de los otros 5 métodos, por lo que probablemente la localización de alguno o algunos de los centroides en esta imagen sea ligeramente errónea.

IMAGEN	Método 1		Método 2		Método 3		Método 4		Método 5		Método 6	
	lin _H	lin _V	lin _H	lin _V	lin _H	lin _V	lin _H	lin _V	lin _H	lin _V	lin _H	lin _V
1	7,12	1,36	1,14	0,85	1,50	0,64	2,00	0,66	1,64	0,85	1,45	1,05
2	7,33	1,37	1,82	1,10	13,28	1,63	6,02	1,28	1,85	2,19	2,13	1,53
3	9,85	1,05	1,96	0,94	8,07	0,93	10,68	1,00	1,82	1,04	2,11	1,07
4	2,08	1,23	0,94	0,91	1,67	0,95	7,93	1,25	1,18	1,21	1,28	1,32
5	3,34	0,77	0,76	0,94	4,07	3,49	3,41	1,09	1,96	0,79	2,55	0,79
6	1,50	1,22	0,71	1,28	1,18	1,36	1,66	1,38	1,44	1,07	3,40	1,14
7	0,86	0,95	0,90	1,04	0,87	0,98	0,84	0,98	0,91	1,02	1,22	1,01
8	2,01	1,42	0,95	1,05	0,96	1,38	0,96	1,43	1,02	1,17	1,32	1,21
9	1,51	1,64	2,95	1,16	6,42	1,24	1,31	1,12	2,11	1,09	1,71	1,07
10	3,22	1,91	1,04	1,21	2,91	2,35	2,17	2,33	1,24	1,33	2,25	1,41
11	4,75	1,07	1,18	2,68	1,30	3,11	5,70	1,30	1,11	1,05	5,12	1,03
12	0,94	1,03	1,29	1,06	1,75	0,88	0,91	1,04	3,11	1,04	2,22	1,03
13	1,85	2,25	1,09	0,88	0,82	1,30	1,45	1,28	1,73	5,76	1,99	3,56
14	1,67	5,33	1,78	1,38	0,60	1,85	0,91	7,02	1,28	1,86	1,10	4,34
15	3,50	0,94	0,71	1,09	0,72	0,68	2,23	0,77	0,85	1,31	2,30	1,06
16	5,46	0,71	1,56	0,95	4,96	1,27	3,20	0,75	1,29	0,90	2,28	1,17
17	2,64	1,24	1,26	1,07	2,47	2,85	5,20	1,68	1,50	2,32	3,17	7,10
18	2,49	2,57	1,00	0,88	18,65	1,82	7,20	1,20	1,57	2,55	1,79	1,21
19	2,03	0,90	1,73	1,06	6,56	0,75	1,21	0,95	1,18	1,03	1,11	1,00
20	2,16	0,98	1,86	0,91	3,92	1,85	1,93	0,95	3,43	0,89	1,89	0,88
21	1,21	0,74	1,15	0,91	1,52	0,89	1,41	0,81	1,07	0,93	1,56	0,91
22	1,98	1,03	5,34	1,02	2,42	1,01	1,34	0,98	2,51	0,98	1,28	1,03
23	0,85	1,24	0,86	1,30	1,07	1,26	0,95	1,16	0,85	1,21	0,84	1,14
24	1,53	0,75	1,66	1,64	1,52	0,78	1,45	0,72	1,55	0,80	1,65	0,77
25	1,84	1,13	5,49	1,16	2,22	1,70	3,21	1,95	4,49	1,75	4,90	2,30
26	1,14	0,91	26,98	0,86	1,96	1,43	5,25	1,09	2,78	0,84	2,48	0,85
27	6,14	1,13	1,22	2,38	0,87	1,01	1,58	1,04	1,14	2,12	1,53	1,72
28	1,50	1,05	2,22	0,88	5,18	0,94	1,78	0,88	2,17	0,85	2,32	0,85
29	1,30	0,86	2,30	0,96	2,12	0,90	1,93	0,88	1,90	0,90	1,99	0,89
30	0,89	1,17	0,95	1,05	0,97	1,13	0,93	1,14	0,91	1,07	0,90	1,08
31	1,58	4,04	3,83	3,24	0,57	1,69	0,83	1,89	4,27	3,05	3,76	2,20
32	3,95	14,92	1,42	1,45	4,22	1,46	3,21	2,76	1,99	1,86	5,05	2,44
33	3,14	1,80	1,73	2,62	2,42	1,28	3,16	1,18	3,52	3,07	5,78	3,26
34	1,90	7,24	8,51	0,96	2,95	2,01	2,42	7,72	3,28	1,60	5,86	2,38
35	4,08	1,50	5,77	1,29	4,20	8,17	4,67	3,06	1,91	1,47	4,67	1,49
Media	2,83	1,98	2,74	1,26	3,34	1,63	2,91	1,62	1,90	1,51	2,48	1,64
>1	31	25	26	22	27	24	28	24	31	25	33	28

Figura 67. Tabla de datos de coeficientes de linealidad

- Se ha calculado también la media de la linealidad en todas las imágenes para cada método. Sin embargo, este parámetro es menos significativo que el número mejorado de imágenes, ya que no sirve de mucho que una imagen tenga un factor de mejora de linealidad muy alto (como ocurre en varios del método 3) en algunas imágenes, si luego falla o es muy bajo en muchas otras. Por lo tanto, interesa que mejore todas, y dentro de la mejora, que sea lo máximo posible, es decir, una solución de compromiso. De nuevo, el método que mejor lo cumple es el 6°.

Como ya se pudo predecir por los resultados visuales de la Figura 66, se corrobora matemáticamente que los métodos 2, 3 y 4, basados en patrones tipo damero, no proporcionan resultados suficientemente buenos para una cámara de baja resolución, mientras que el método 1 da buenos resultados de corrección de distorsión (aunque no mejores que el sexto), y que los que mejor funcionan son los métodos 5 y 6, basados en calibración a través de los patrones circulares con búsqueda de centroides sub-pixélica. Dentro de estos dos métodos, el que mejor funciona es el sexto, debido a que los círculos impresos sobre el patrón tienen un radio mayor, lo que ayuda a una búsqueda del centroide más precisa en cámaras de baja resolución.

5. Calibración estéreo de cámaras ToF

En el presente trabajo no es necesario recurrir a la visión estéreo para obtener información de profundidad, ya que estos datos en teoría son proporcionados por las cámaras de tiempo de vuelo. El hecho de estudiar la configuración estéreo proporciona la posibilidad de poder fusionar los datos de dos o más cámaras ToF que capturen imágenes de una misma escena, pero colocadas en distintos puntos. Para ello, se requiere conocer la posición relativa entre ambas, y de esta manera poder trasladar y rotar una de las nubes de puntos para que su origen de coordenadas coincida con la de la otra cámara. Por lo tanto, la utilidad que va a tener para este proyecto la geometría estéreo es realizar una calibración para calcular dichas matrices de rotación y traslación que permitan el paso entre distintas cámaras.

Cuando se trabaja con cámaras 2D que obtienen una imagen de la escena, se están proyectando puntos tridimensionales en puntos bidimensionales. Como se observa en la Figura 68, dos puntos 3D que estén alineados con respecto al centro de proyección o centro óptico, como son P y Q, van a tener la misma proyección en 2D sobre el plano imagen. Es decir, P' y Q' van a tener las mismas coordenadas (u, v) . Esto significa que en la imagen 2D de la cámara se pierde la dimensión de la profundidad.

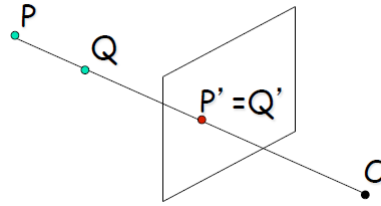


Figura 68. Proyección de dos puntos 3D (P y Q) en 2D (P' y Q')¹⁰

De manera matemática [9], se puede calcular la proyección 2D de un punto \tilde{p} si se conoce la matriz $C = [KR|KRT]$ que contiene todos los parámetros de calibración de la cámara (extrínsecos e intrínsecos) según la expresión (29):

$$\tilde{p} = C \cdot \tilde{P} \rightarrow \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = C \cdot \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (29)$$

Sabiendo que la matriz C está compuesta por tres filas que se pueden llamar r_1^C, r_2^C y r_3^C , se puede reescribir la ecuación (29) de la siguiente forma:

$$\begin{aligned} u \cdot (r_3^C \cdot P) - r_1^C \cdot P &= 0 \\ v \cdot (r_3^C \cdot P) - r_2^C \cdot P &= 0 \end{aligned} \quad (30)$$

Como se observa en la ecuación (30), de cada cámara se pueden obtener 2 ecuaciones lineales. Sin embargo, en el sistema existen 3 incógnitas (X, Y, Z), por lo que no es suficiente para resolver el sistema. Este es el razonamiento matemático por el cual no se puede calcular profundidad solamente con una cámara.

¹⁰ <http://cs.gmu.edu/~kosecka/cs482/lect-stereo-motion.pdf>

Para poder solventar este problema y obtener datos de la profundidad de los puntos de la escena, se puede recurrir a la visión estéreo, que consiste en utilizar dos cámaras que tomen las mismas imágenes de la escena, pero desde dos posiciones distintas. Cada cámara aportaría 2 ecuaciones lineales al sistema (como las de la ecuación (30)), de forma que habrá 4 ecuaciones y ya se podría resolver la tercera incógnita y obtener profundidad. Por ejemplo, el ser humano utiliza un sistema estéreo para la visión, en la que cada ojo se correspondería con una cámara.

Como el sistema de reconstrucción montado consta de 4 cámaras en posiciones fijas sobre las cuatro esquinas de un rectángulo (Figura 1), se va a realizar una aproximación estéreo de las cámaras contiguas dos a dos, hasta cerrar el rectángulo. Por lo tanto, se considerará el estudio y caracterización de cuatro configuraciones estéreo, mediante los pares de cámaras 1-2, 2-3, 3-4 y 4-1, que permitirían enlazarlas entre sí para obtener los parámetros del sistema completo formado por las cuatro cámaras. De hecho, solamente sería estrictamente necesario realizar 3 de las 4 posibles calibraciones estéreo de los pares de cámaras citados, ya que con ello sería suficiente para cerrar el sistema. Es decir, los parámetros del par de cámaras 4-1 se podría obtener a partir de la combinación de las otras 3 calibraciones: $C_{41} = C_{12}C_{23}C_{34}$, donde C_{ij} indica los parámetros de cada calibración estéreo de cada par de cámaras de su subíndice. Además, se sabe $C_{12}C_{23}C_{34}C_{41} = 0$, es decir, que al aplicar las 4 rotaciones y traslaciones de los pares de calibración, se volvería a la posición inicial.

5.1. Sistema estéreo. Geometría epipolar.

La visión estéreo se basa en el proceso de recuperar la profundidad a partir de imágenes obtenidas por cámaras desde dos (o más) puntos de vista de la misma escena. Los sistemas estéreo que se van a modelar en el presente trabajo están compuestos por dos cámaras (izquierda y derecha) colocadas en posiciones fijas, desde las que se adquieren imágenes de la escena común a ambas, como se muestra en la Figura 69. En esta figura aparecen las dos cámaras separadas una distancia b (línea base) entre sí, teniendo cada una su sistema de referencia propio de coordenadas y su correspondiente plano imagen sobre el que se proyecta el mismo punto 3D $P = (X, Y, Z)$.

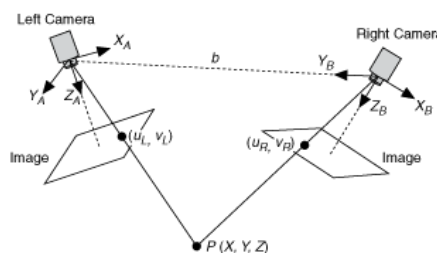


Figura 69. Sistema estéreo con cámara izquierda y derecha¹¹

Un sistema estéreo dependerá de los parámetros internos de cada una de las cámaras (distancia focal, centro óptico...) que caracterizan la transformación o el paso de coordenadas del mundo (3D) a píxeles del sistema de coordenadas de cada cámara (2D), así como de la posición relativa y la orientación de las cámaras entre sí (matrix de rotación R y vector de traslación T), de manera que se pueden aplicar las siguientes expresiones para referir los puntos de una de las cámaras a la otra dado un par estéreo cualquiera con una cámara derecha y otra izquierda:

¹¹ <http://zone.ni.com/reference/en-XX/help/372916M-01/nivisionconceptsdata/guid-cb42607e-f256-40f5-ab6e-28ec3a33bcd/>

$$\begin{aligned} P_R &= R \cdot (P_L - T) \\ P_L &= R \cdot P_R + T \end{aligned} \quad (31)$$

5.1.1. Vector de traslación y matriz de rotación. Ángulos de Euler.

Como se ha indicado, cuando se trabaja con datos referidos a dos sistemas de coordenadas distintos, como van a ser las nubes de puntos correspondientes a cada una de las cámaras de tiempo de vuelo de manera independiente, y se quiere fusionar dichos datos obtenidos, es necesario referir la información de ambos sistemas a un único sistema de referencia común. Para ello, se puede aplicar una rotación y una traslación a uno de los conjuntos de datos aplicando la ecuación (31), consiguiendo de esta manera que los datos se refieran a un mismo origen.

En primer lugar, el vector de traslación T sería un vector tridimensional con componentes $T = (T_x, T_y, T_z)^T$ que se calcularía simplemente cómo la diferencia entre los centros ópticos de cada uno de los sistemas de coordenadas:

$$T = O_R - O_L \quad (32)$$

Para hallar la matriz de rotación R es necesario conocer los ángulos que caracterizan la rotación tridimensional existente. Para ello, podemos basarnos en los ángulos de Euler α , β y γ , que aparecen en la Figura 70. Los ángulos de Euler [20] describen una secuencia de 3 rotaciones de los ejes de un sistema de referencia móvil XYZ sobre los ejes de un sistema de referencia fijo xyz .

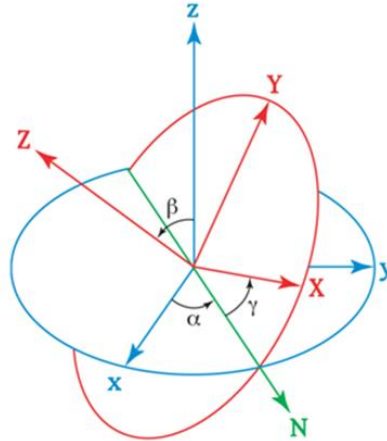


Figura 70. Ángulos de Euler

Para indicar con qué se corresponde cada ángulo en la Figura 70 se ha de introducir un eje adicional llamado ‘línea de nodos’ N , que es la intersección entre los planos xy y XY . Según esto, los ángulos de Euler serán:

- α : ángulo que forman el eje N con el eje x .
- β : ángulo que forman el eje N con el eje X .
- γ : ángulo que forman entre sí los ejes z y Z .

Sin embargo, lo que interesa calcular es la matriz de rotación R que va a permitir realizar el cambio de coordenadas de un sistema a otro a partir de estos ángulos. Ésta va a ser una matriz de dimensiones 3×3 como la de la ecuación (33).

$$R = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \quad (33)$$

Cada ángulo de Euler α , β y γ se corresponde con una rotación simple. Cada una de estas rotaciones se pueden expresar según las siguientes matrices, respectivamente:

$$R_\alpha = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{pmatrix}, R_\beta = \begin{pmatrix} \cos \beta & 0 & -\sin \beta \\ 0 & 1 & 0 \\ \sin \beta & 0 & \cos \beta \end{pmatrix}, R_\gamma = \begin{pmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (34)$$

A continuación, para realizar la rotación, se multiplican las matrices de la expresión (111) en el orden determinado, obteniendo finalmente la expresión (35):

$$R = R_\gamma R_\beta R_\alpha \quad (35)$$

$$R = \begin{pmatrix} \cos \gamma \cos \beta & \cos \gamma \sin \beta \sin \alpha - \sin \gamma \cos \alpha & \cos \gamma \sin \beta \cos \alpha + \sin \gamma \sin \alpha \\ \sin \gamma \cos \beta & \sin \gamma \sin \beta \sin \alpha + \cos \gamma \cos \alpha & \sin \gamma \sin \beta \cos \alpha - \cos \gamma \sin \alpha \\ -\sin \beta & \cos \beta \sin \alpha & \cos \beta \cos \alpha \end{pmatrix}$$

Para llevar a cabo el proceso inverso, es decir, obtener los ángulos a partir de una matriz de rotación R dada, se aplican las siguientes ecuaciones:

$$\beta = \text{atan2}(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2}) \quad (36)$$

$$\gamma = \text{atan2}\left(\frac{r_{21}}{\cos \beta}, \frac{r_{11}}{\cos \beta}\right) \quad (37)$$

$$\alpha = \text{atan2}\left(\frac{r_{32}}{\cos \beta}, \frac{r_{33}}{\cos \beta}\right) \quad (38)$$

Sin embargo, los ángulos que devuelve la toolbox de calibración de MATLAB® en el vector que llama *om* por defecto no son los ángulos de Euler (los de la expresión (111)). A pesar de que la matriz de rotación R que devuelve va a tener finalmente el mismo valor, la manera de calcularla no es la misma, ya que los ángulos de los que se parte son otros. Para ello, MATLAB® hace uso del algoritmo ‘rodrigues’, que consiste en calcular la matriz R a partir de la siguiente fórmula:

$$R = I + [n]_X \cdot \sin \theta + [n]_X^2 (1 - \cos \theta) \quad (39)$$

Donde I es la matriz identidad, θ es el ángulo en cuestión de rotación en radianes y $[n]_X$ es un vector simétrico normalizado del vector del eje de rotación. En el apartado de calibración estéreo de MATLAB® se detalla más información acerca del cálculo de la matriz de rotación mediante ‘Rodrigues’.

5.2. Sistema de calibración

La calibración de un sistema estéreo consiste en la determinación de la relación matemática existente entre las coordenadas tridimensionales de los puntos de una escena del mundo real y sus correspondientes proyecciones en el plano imagen de cada una de las cámaras. El objetivo principal de este método va a ser hallar la matriz de rotación R y el vector de traslación T que modelan la posición relativa del sistema de referencia de las dos cámaras

estéreo, de manera que se puedan hacer coincidir sus nubes de puntos 3D mediante estos parámetros hallados en la calibración.

Para realizar la calibración estéreo se va a necesitar adquirir varias imágenes de la misma escena desde la perspectiva de cada una de las cámaras, y a continuación, tomar los parámetros de calibración de cada una de ellas y sus puntos correspondientes en cada par de imágenes para ejecutar el algoritmo de calibración estéreo en MATLAB® que devolverá los parámetros extrínsecos de la calibración, así como una optimización de la calibración previa de cada una de las cámaras del sistema estéreo. Se detalla a continuación el proceso.

5.2.1. Obtención de imágenes estéreo y puntos característicos

En primer lugar, es necesario tomar imágenes de la escena desde la perspectiva de cada una de las cámaras y encontrar los puntos correspondientes entre los pares de imágenes capturados. Podrían hallarse estos puntos mediante geometría epipolar (referirse al ANEXO III. Geometría epipolar), pero no va a ser necesario ya que se puede obtener esta correspondencia de manera más sencilla aprovechando el algoritmo de búsqueda de centroides de un patrón. Es decir, se colocará de nuevo el patrón en varias posiciones de la escena, de la misma manera que se hizo para la calibración de las cámaras de manera individual, y se tomarán imágenes de la escena desde ambas cámaras. De esta forma, si se capturan n posiciones del patrón, existirán n pares de imágenes, $2n$ imágenes en total, n para cada cámara, sobre las que habrá que calcular los centroides con precisión sub-pixélica. En la Figura 71 se muestran dos pares de imágenes tomadas con el patrón en distintas posiciones, y además, se ha utilizado el método explicado en el proyecto para hallar la posición de los centroides.



Figura 71. Imágenes estéreo de la cámara 1 y 2 con los centroides marcados.

Por lo tanto, el primer paso va a ser adquirir imágenes desde ambas cámaras y ejecutar el algoritmo de búsqueda de centroides. Esto, además de devolver las posiciones características de los puntos de control en cada imagen, permitirá realizar una calibración inicial de cada una de las dos cámaras de manera independiente, lo que será necesario para comenzar la calibración estéreo (apartado 5.2.3).

5.2.2. Algoritmo ‘Rodrigues’

Antes de comenzar con la explicación del algoritmo de calibración de MATLAB®, es importante comprender que los tres ángulos que proporciona no se corresponden con los ángulos de Euler, a pesar de que la matriz de rotación R sí sea la misma que se obtiene a partir de ellos. Esto es debido a que utiliza una función basada en el método ‘Rodrigues’, que calcula R según la expresión (39).

Mientras que los ángulos de Euler realizan rotaciones bidimensionales de los planos xy , yz y xz , Rodrigues [21][22] se utiliza para llevar a cabo rotaciones de puntos tridimensionales alrededor de un eje dado un determinado ángulo, siendo el resultado final, la matriz de rotación

R , el mismo para ambos casos. En la Figura 72 se muestra el proceso geométrico que se realiza al realizar una rotación mediante este método. Es importante conocer esto para no pensar que los ángulos que devuelve la toolbox son los de Euler al exportarlos para utilizarlos con código propio u otras aplicaciones externas y precipitarse en decir que el proceso de calibración estéreo no ha sido correcto.

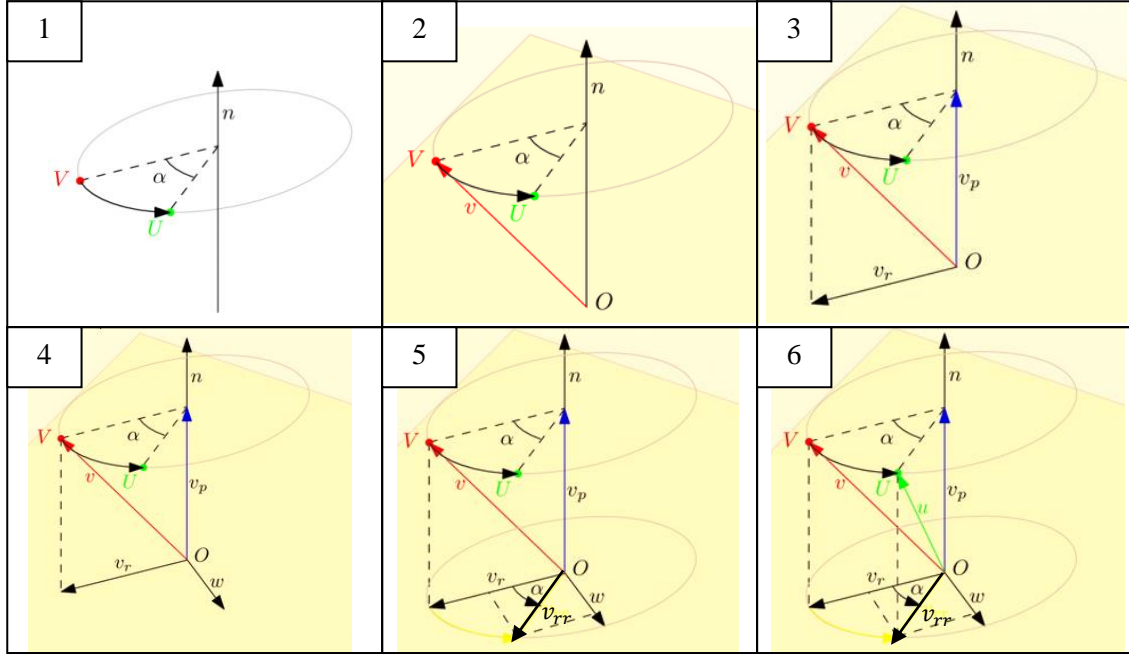


Figura 72. Operaciones realizadas por el método Rodrigues¹²

En el primer paso, se tiene un punto V que se desea rotar un ángulo α en torno al eje n . El resultado de esta rotación finalmente deberá ser el punto U . Para ello, en el segundo paso se define un vector v que va desde el origen O hasta el punto V y un plano en el origen perpendicular al eje de rotación n . También se obtiene el vector normalizado del eje de rotación n , $|n| = 1$. A continuación, se proyecta el vector v sobre el plano (v_r) y sobre el eje n (v_p) según las siguientes expresiones:

$$v_p = (n \cdot v)n \quad (40)$$

$$v_r = v - v_p \quad (41)$$

En el paso 4, se crea un vector ω perpendicular al plano formado por los vectores v y v_p mediante el siguiente producto vectorial:

$$\omega = n \times v \quad (42)$$

En el paso 5, se realiza una rotación 2D del vector v_r sobre el plano v_r , ω mediante la expresión (43), obteniendo un nuevo vector v_{rr} :

$$v_{rr} = v_r \cdot \cos \alpha + \omega \sin \alpha \quad (43)$$

Por último, hay que pasar del plano en el origen en el que está el vector v_{rr} al punto U . Para ello, solamente se suman los vectores v_{rr} y v_p :

$$u = v_{rr} + v_p \quad (44)$$

Sustituyendo las operaciones realizadas desde la (40) hasta la (43) sobre la expresión (44), se obtiene:

¹² <http://electroncastle.com/wp/?p=39>

$$\begin{aligned}
u &= (v - (n \cdot v)n) \cos \alpha + (n \times v) \sin \alpha + (n \cdot v)n \\
u &= v \cos \alpha - (n \cdot v)n \cos \alpha + (n \times v) \sin \alpha + (n \cdot v)n \\
u &= v \cos \alpha - (n \cdot v)n(1 - \cos \alpha) + (n \times v) \sin \alpha
\end{aligned} \tag{45}$$

Se puede simplificar la ecuación final (45) convirtiendo los productos de vectores en simples multiplicaciones. Asumiendo que todos los vectores son vectores columna, el producto \cdot puede ser calculado según la expresión:

$$v_p = (v \cdot n)n = (n^T v)n = nn^T v \tag{46}$$

Por otro lado el producto cruzado “x” se puede calcular utilizando el método de la matriz antisimétrica. El vector n se puede expresar en la forma de su matriz antisimétrica de la siguiente forma:

$$[n]_x = \begin{pmatrix} 0 & -n_3 & n_2 \\ n_3 & 0 & -n_1 \\ -n_2 & n_1 & 0 \end{pmatrix} \tag{47}$$

Y el producto vectorial de la ecuación (42) se puede reescribir como la siguiente multiplicación:

$$\omega = n \times v = [n]_x v \tag{48}$$

Ahora, sustituyendo las ecuaciones (47) y (48) en la expresión (45) se obtiene:

$$u = v \cos \alpha - nn^T v(1 - \cos \alpha) + [n]_x v \sin \alpha \tag{49}$$

Por último, para obtener la matriz de rotación R , hacemos uso de la expresión aplicada para conseguir rotar el punto V :

$$u = Rv \tag{50}$$

$$u = (\cos \alpha - nn^T(1 - \cos \alpha) + [n]_x \sin \alpha)v \tag{51}$$

Despejando finalmente, la matriz de rotación R es:

$$R = \cos \alpha - nn^T(1 - \cos \alpha) + [n]_x \sin \alpha \tag{52}$$

Esta expresión puede aún simplificarse más sabiendo que $nn^T = [n]_x^2 + I$, siendo I la matriz identidad, y sustituyendo y operando, se obtiene la expresión final de R mediante el método Rodrigues:

$$R = I + [n]_x \cdot \sin \alpha + [n]_x^2(1 - \cos \alpha) \tag{53}$$

Por lo tanto, este es el procedimiento que utiliza MATLAB® para calcular la matriz de rotación a partir de los vectores de rotación de la matriz *om* que devuelve el proceso de calibración. Es un paso muy importante a tener en cuenta, porque si se pretende comparar con otras calibraciones, normalmente se utilizan los ángulos de Euler o el pan, tilt and roll, y podría sacarse una conclusión precipitada de que el resultado que devuelve MATLAB® es incorrecto.

5.2.3. Calibración mediante MATLAB®

Para realizar la calibración estéreo se van a utilizar las funciones disponibles de la toolbox de MATLAB® estéreo de Yves-Bouquet, que presenta la apariencia de la Figura 73.



Figura 73. Toolbox de calibración estéreo

El primer paso que se ha de realizar es calibrar cada una de las cámaras de manera independiente y almacenar el fichero *.mat* que contiene los datos de calibración de cada una de ellas en un mismo directorio, entre los que se encuentran también las coordenadas de los centroides obtenidos en cada una de las imágenes. Primero, se cargan cada uno de estos ficheros, indicando cuál de ellos corresponde a cada una de las cámaras, y se imprime por pantalla en MATLAB® una aproximación inicial de los parámetros intrínsecos de cada una de ellas (de su calibración) y de las matrices R y T . Hay que tener en cuenta que las matrices que proporciona son la R y T de la cámara derecha sobre la cámara izquierda. Por otro lado, es muy importante que los pares de imágenes estéreo de la cámara derecha e izquierda se correspondan entre sí a la hora de hallar los centroides e indicar por el nombre de la imagen que ambas se corresponden entre sí, porque si no los datos no serán consistentes y no se podrá realizar la calibración. Por lo tanto, hay que tener mucho cuidado con el orden de almacenamiento de las imágenes y de sus centroides.

El algoritmo que realiza MATLAB® para este cálculo inicial puede resumirse en los pasos que se muestran en el esquema de la Figura 74:

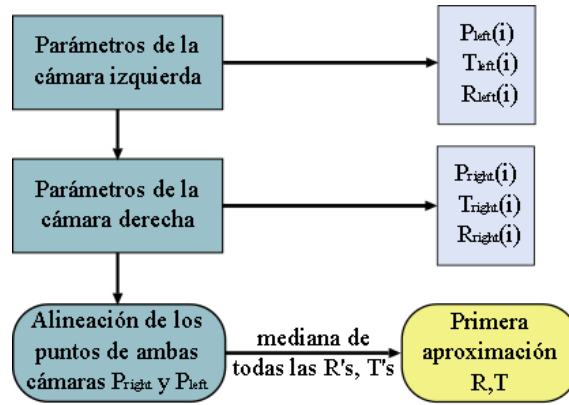


Figura 74. Proceso inicial de calibración

A continuación se explican cada uno de los pasos nombrados en la Figura 74 de manera más detallada.

- Carga los datos de calibración de la cámara izquierda y comienza un bucle con tantas iteraciones como número de imágenes i existan, en el cual realiza la operación de la expresión (54), donde $P(i)$ son las coordenadas sobre el plano de los centroides del patrón, las cuales son rotadas y trasladadas según $R_c(i)$ y $T_c(i)$, que son la matriz de rotación y el vector de traslación que definen la posición 3D del patrón en la escena, que fueron calculadas durante la calibración de la cámara izquierda. De esta manera, se obtiene $P_c(i)$, que son las coordenadas tridimensionales de los puntos del patrón referidas al sistema de coordenadas de la cámara izquierda.

$$P_{c_{left}}(i) = R_{c_{left}}(i)P_{left}(i) + T_{c_{left}}(i) \quad (54)$$

Se almacenan en un array todos los puntos $P_{c_{left}}(i)$ de cada una de las i imágenes, así como los vectores $om_{c_{left}}(i)$ ($om = rodrigues(R)$) y $T_{c_{left}}(i)$.

- Se realiza el mismo proceso para la cámara derecha, obteniendo sus vectores correspondientes $P_{c_{right}}$, $om_{c_{right}}$ y $T_{c_{right}}$.

$$P_{c_{right}}(i) = R_{c_{right}}(i)P_{right}(i) + T_{c_{right}}(i) \quad (55)$$

- Tras obtener los vectores $R(om)$, T y los puntos 3D de los mismos centroides referidos a cada cámara en cada imagen, se alinea la estructura del patrón a la primera vista (la de la cámara izquierda) aplicando las siguientes ecuaciones para cada imagen del patrón:

$$R_{ref}(i) = R_{right}(i) \cdot R_{left}(i) \quad (56)$$

$$T_{ref}(i) = T_{right}(i) - R_{ref}(i) \cdot T_{left}(i) \quad (57)$$

- Finalmente, se hace la mediana de todos los $om_{ref}(i)$ y $T_{ref}(i)$ almacenados, y la solución que proporcione esta mediana será la aproximación inicial de los parámetros extrínsecos del sistema estéreo.

A continuación, ha de llevarse a cabo un proceso de optimización global de la calibración estéreo utilizando la función ‘*go_calib_stereo.m*’, en la que se lleva a cabo un proceso iterativo con el objetivo de minimizar los errores de los parámetros, partiendo de la aproximación inicial de éstos, de manera similar a la calibración de una sola cámara. En este paso se puede elegir si se desea optimizar también los parámetros de la calibración independiente de cada una de las cámaras o si se prefiere conservar los resultados obtenidos.

Una vez que se hayan obtenido los datos de calibración, se puede visualizar mediante la toolbox una gráfica con la posición relativa de las cámaras entre sí y los puntos de los patrones en 3D con respecto a éstas, tal y como se muestra en la Figura 75.

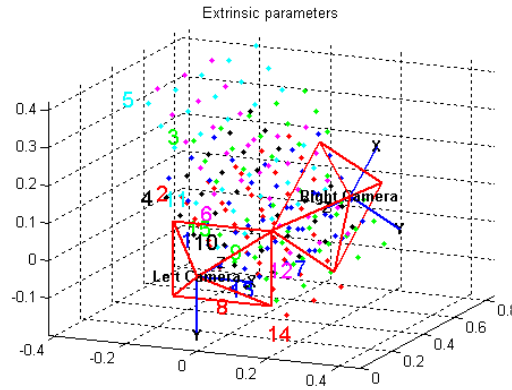


Figura 75. Posición relativa entre cámaras 1 y 2 y los centroides usados para calibrar

En la Figura 76 se muestran la representación de la posición relativa de los sistemas de referencia de cada par de cámaras utilizado en el presente trabajo, obtenido a partir de las 4 calibraciones estéreas realizadas.

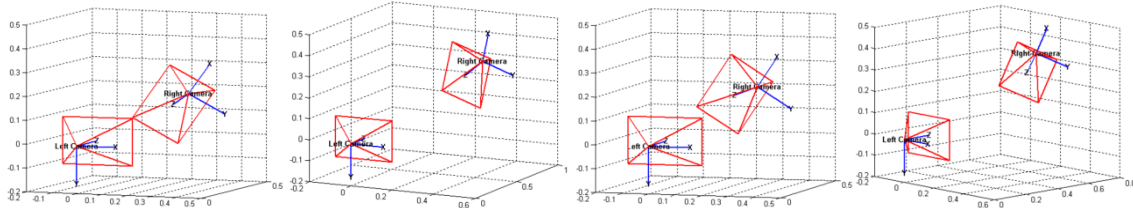


Figura 76. Sistemas de referencia de cada par de cámaras tras la calibración estéreo. De izquierda a derecha, pares de cámaras 1-2, 2-3, 2-4 y 4-5

5.3. Reconstrucción mediante triangulación

Una manera de comprobar si la calibración estéreo ha sido realizada con éxito es re-proyectar los puntos de los centroides a las coordenadas tridimensionales del mundo a partir de sus coordenadas 2D en cada una de las cámaras y los datos de calibración obtenidos. Si el proceso ha sido realizado de manera apropiada, las distancias existentes entre los puntos característicos del patrón serán cercanas a las reales.

En primer lugar, se van a expresar las matrices de calibración correspondientes a cada una de las dos cámaras que forman el sistema estéreo, según las ecuaciones (58) y (59), sabiendo que el sistema al que se van a referenciar los puntos es con respecto a la cámara de la izquierda, y que las matrices K_R y K_L son las correspondientes matrices que contienen los parámetros intrínsecos de cada una de las cámaras.

$$p_R = C_R P \rightarrow C_R = [K_R R | K_R RT] = [M_R | M_R T] = \begin{pmatrix} m_{11R} & m_{12R} & m_{13R} & mt_{1R} \\ m_{21R} & m_{22R} & m_{23R} & mt_{2R} \\ m_{31R} & m_{32R} & m_{33R} & mt_{3R} \end{pmatrix} \quad (58)$$

$$p_L = C_L P \rightarrow C_L = [K_L | 0] = \begin{pmatrix} m_{11L} & m_{12L} & m_{13L} & 0 \\ m_{21L} & m_{22L} & m_{23L} & 0 \\ m_{31L} & m_{32L} & m_{33L} & 0 \end{pmatrix} \quad (59)$$

A partir de estos parámetros se construye el siguiente sistema de ecuaciones:

$$\begin{cases} p_R = C_R P \rightarrow (u_L & v_L & 1)^T = C_L (X & Y & Z & 1)^T \\ p_L = C_L P \rightarrow (u_R & v_R & 1)^T = C_R (X & Y & Z & 1)^T \end{cases} \quad (60)$$

Si se llama a cada fila de las matrices C_L y C_R $r_i^{C_R}$ y $r_i^{C_L}$ respectivamente, la ecuación (60) se puede reescribir como se muestra a continuación:

$$\begin{cases} u_L(r_3^{C_L} \cdot P) - r_1^{C_L} \cdot P = 0 \\ v_L(r_3^{C_L} \cdot P) - r_2^{C_L} \cdot P = 0 \\ u_R(r_3^{C_R} \cdot P) - r_1^{C_R} \cdot P = 0 \\ v_R(r_3^{C_R} \cdot P) - r_2^{C_R} \cdot P = 0 \end{cases} \quad (61)$$

La incógnita a despejar es el punto tridimensional P en un sistema del tipo:

$$A \cdot x = b \quad (62)$$

De esta manera se pueden re-proyectar los puntos del patrón a través de sus coordenadas bidimensionales en cada una de las imágenes estéreo, y de esta manera extraer conclusiones de la eficiencia de la calibración.

Un aspecto a tener en cuenta es que la matriz de rotación R que debe ser sustituida en estas ecuaciones no es la que la toolbox proporciona directamente en la calibración. La que devuelve el proceso de calibración sirve para trabajar en unidades de píxeles. Sin embargo, si se quiere realizar rotaciones con unidades del mundo real, es decir, en el mundo tridimensional y en el sistema métrico, debe utilizarse la matriz inversa de R .

5.4. Comprobación de la calibración de los 4 pares estéreo de cámaras ToF

Como finalmente se va a tratar de lograr una reconstrucción 3D de la escena vista por las 4 cámaras, un paso inicial es comprobar que las calibraciones de los 4 sistema estéreo formados por cada par de cámaras contiguas han sido realizadas correctamente.

En primer lugar se va a describir el proceso para el par estéreo de las cámaras 1 y 2. Para ello, se va a tomar un set de 4 imágenes capturadas con dichas cámaras y se va a comprobar la precisión de la calibración realizada. En la Figura 77 se muestran estas imágenes, y cada una está recuadrada con un color. Se toman los centroides obtenidos de cada una de esas imágenes junto con los parámetros de calibración del sistema estéreo, y se resuelve el sistema que resulta de la ecuación (61). Es importante que los centroides que se desea reconstruir en cada imagen en este paso tengan la distorsión previamente corregida, ya que el modelo de reconstrucción 3D explicado no contempla dicho efecto, solamente se corresponde con el modelo pinhole. De esta manera, se obtienen los puntos 3D de cada centroide, y se representan en la Figura 78. El color de representación de cada patrón se corresponde con el color asociado a cada par de imágenes estéreo de la Figura 77.

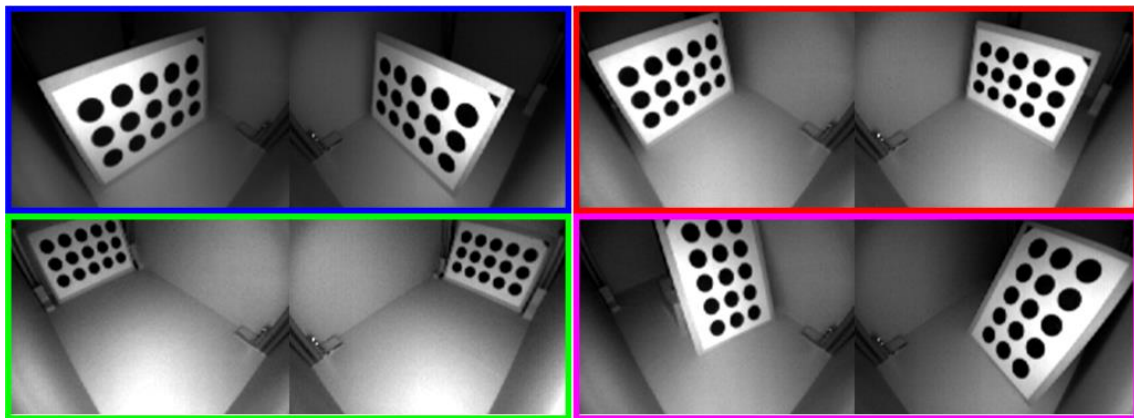


Figura 77. Imágenes estéreo del patrón en distintas posiciones de la escena con las cámaras 1 y 2

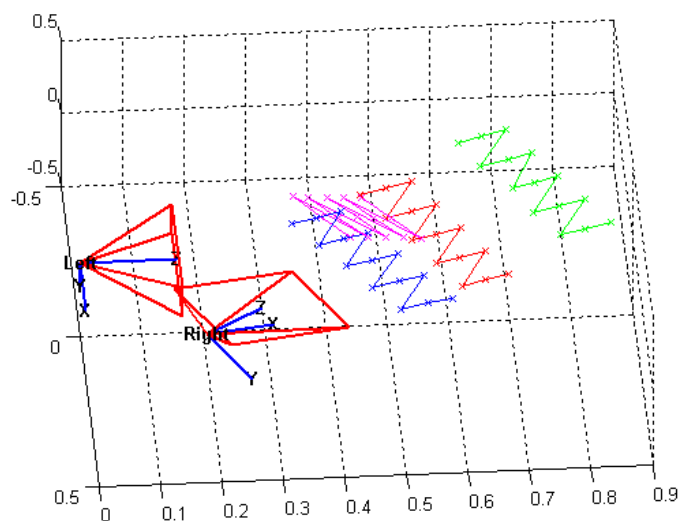


Figura 78. Reconstrucción 3D de los centroides de los patrones de cada una de las imágenes de la Figura 77

A priori, según los resultados de la gráfica 3D de la Figura 78, los centroides se re-proyectan correctamente, dando lugar a la forma del patrón. Además, se observa que los patrones azul, rojo y verde tienen su lado ‘largo’ paralelo al eje que une ambas cámaras, mientras que el patrón rosa es perpendicular a este. Por otro lado, se guardan también relaciones de distancias entre patrones, ya que de los 3 patrones que están apoyados con su lado largo en el suelo, en las imágenes de escala de grises el más cercano es el azul, a continuación el rojo, y por último el verde, de la misma manera que ocurre en la reconstrucción 3D. Por tanto, se puede concluir que la reconstrucción parece correcta, pero hay que comprobar si el factor de escala aplicado también lo es. Es decir, si las distancias se corresponden con las reales. Para esto, se van a calcular distancias existentes entre los centroides re-proyectados en 3D, sabiendo que la distancia ideal entre ellos ha de ser 7.9 cm. En la tabla de la Figura 79 se muestran una serie de medidas realizadas para cada uno de los cuatro patrones, todas ellas expresadas en cm. En la primera fila de cada par de centroides entre los que se está midiendo la distancia se muestra dicha magnitud, y en la segunda fila, el error con respecto a la medida ideal en valor absoluto y en tanto por ciento, es decir, el valor del error relativo.

Centroides	Real	Azul		Rojo		Verde		Rosa	
1-2	7.9	7.91		7.99		7.99		8.03	
		0.01	0.12%	0.09	1.1%	0.09	1.1%	0.18	1.64%
1-4	7.9	8.05		8.05		8.05		7.95	
		0.15	1.8%	0.15	1.8%	0.15	1.8%	0.05	0.63%
1-13	31.6	31.84		32.2		32.2		31.83	
		0.24	0.75%	0.6	1.9%	0.6	1.9%	0.23	0.72%
1-15	35.32	35.79		36.03		36.03		35.79	
		0.47	1.33%	0.71	2%	0.71	2%	0.47	1.33%

Figura 79. Distancias (en centímetros) en los centroides re-proyectados mediante los datos de calibración estéreo de las cámaras 1-2

Como se puede comprobar según los datos de la Figura 79, los errores en la medida son bastante bajos, no llegando a superar en ningún caso un error relativo más del 2%. Por lo tanto, se puede concluir que las cámaras 1 y 2 se han calibrado correctamente y se obtienen medidas tridimensionales que se ajustan a las del mundo real.

A continuación se van a mostrar los mismos resultados de otro de los pares de cámaras, las cámaras 2 y 3, en los que se han elegido otros 4 patrones de sus imágenes capturadas, tal y como se muestra en la Figura 80. Como se observa en las imágenes, la línea base entre ambas cámaras, es decir, la distancia que las separa, es bastante mayor que en el caso de las cámaras 1 y 2, lo que restringe en gran medida las posiciones y orientaciones en las que se puede colocar el patrón en la escena para tomar imágenes de manera que el patrón se vea desde ambas cámaras.

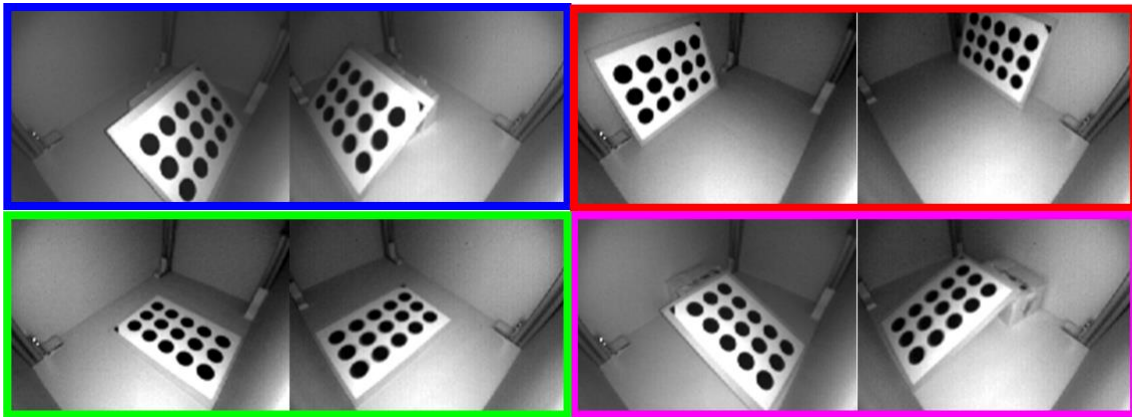


Figura 80. Imágenes estéreo del patrón en distintas posiciones de la escena con las cámaras 2 y 3

En la Figura 81 se muestra la representación 3D de cada uno de los patrones de las imágenes estéreo capturadas. De nuevo, se puede concluir que los patrones calculados guardan relación de posición y forma tal y como aparecen en las imágenes. En la Figura 82 se calculan los errores de medida entre dichos centroides 3D, de la misma manera que se realizó para el par de cámaras 1-2. De nuevo, el error continúa manteniéndose en unos límites aceptables, por debajo del 2%, por lo que se puede concluir que también para este caso, la calibración estéreo llevada a cabo es válida.

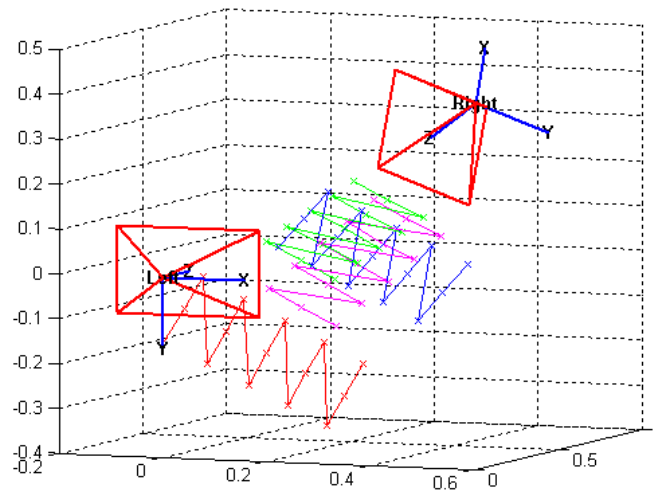


Figura 81. Reconstrucción 3D de los centroides de los patrones de cada una de las imágenes de la Figura 80

Centroides	Real	Azul		Rojo		Verde		Rosa	
1-2	7.9	8.04		7.9		7.78		7.87	
		0.14	1.77%	0	0%	-0.12	1.51%	-0.03	0.37%
1-4	7.9	7.76		7.83		7.78		7.87	
		-0.14	1.77%	-0.07	0.88%	-0.12	1.51%	-0.03	0.37%
1-13	31.6	31.38		31.08		31.88		31.62	
		-0.22	0.69%	-0.52	1.6%	0.28	0.88%	0.02	0.06%
1-15	35.32	35.19		34.98		35.55		35.23	
		-0.13	0.36%	-0.34	0.92%	0.23	0.65%	0.47	0.25%

Figura 82. Distancias (en centímetros) en los centroides re-proyectados mediante los datos de calibración estéreo de las cámaras 2-3

Por último, para los pares de cámaras que restan (3-4 y 4-1) se van a tomar otras 4 imágenes estéreo del patrón en distintas posiciones y se va a mostrar solamente los resultados finales cuantitativos, que son las tablas de error de medida de los centroides tridimensionales, para poder estudiar la validez de sus datos de calibración y afirmar que el sistema completo ha sido calibrado.

Centroides	Real	1		2		3		4	
1-2	7.9	7.73		7.75		7.94		7.91	
		0.17	2.15%	-0.15	1.89%	0.04	0.5%	0.01	0.12%
1-4	7.9	7.83		7.71		7.78		7.75	
		-0.07	0.88%	-0.19	2.4%	-0.12	1.51%	-0.15	1.89%
1-13	31.6	32.26		31.64		31.7		30.9	
		0.66	2.08%	0.04	0.12%	0.1	0.31%	-0.7	2.2%
1-15	35.32	36.16		35.12		34.92		35.41	
		0.84	2.3%	-0.2	0.56%	-0.4	1.13%	0.09	0.25%

Figura 83. Distancias (en centímetros) en los centroides re-proyectados mediante los datos de calibración estéreo de las cámaras 3-4

Centroides	Real	1		2		3		4	
1-2	7.9	7.9		8.01		7.8		7.84	
		0	0%	0.11	1.39%	-0.1	1.26%	-0.06	0.75%
1-4	7.9	7.77		8.01		7.77		7.93	
		-0.13	1.64%	0.11	1.39%	-0.13	1.64%	0.03	0.37%
1-13	31.6	31.67		31.85		31.3		31.63	
		0.07	0.22%	0.25	0.79%	-0.3	0.94%	0.03	0.09%
1-15	35.32	34.82		34.95		35.18		35.07	
		-0.5	1.41%	-0.37	1.04%	-0.14	0.39%	-0.25	0.7%

Figura 84. Distancias (en centímetros) en los centroides re-proyectados mediante los datos de calibración estéreo de las cámaras 4-1

Los resultados de las 4 calibraciones estéreo presentan un error que se encuentra dentro de los márgenes aceptables, por lo que se puede concluir que las calibraciones han sido correctas y se pueden utilizar los datos obtenidos para la reconstrucción 3D.

5.5. Relación de los 4 pares de calibración estéreo

Una vez que se han obtenidos los datos de las calibraciones estéreo de los 4 pares de cámaras existentes en el sistema, es necesario unificar todas ellas a un único sistema de referencia, y se ha elegido el sistema de coordenadas de la cámara número 1. Por lo tanto, las cámaras 1 y 2 ya están referenciadas a dicho origen mediante la calibración estéreo del par 1-2, pero la cámara 3 tendrá que relacionarse a dicho sistema a través de la calibración 2-3, y de la misma manera se procederá con la cámara número 4. Yendo paso a paso, en primer lugar referenciamos la cámara 2 al sistema de la cámara 1 mediante la expresión (63), sabiendo que R_{12} y T_{12} son los parámetros extrínsecos de rotación y traslación obtenidos del proceso de calibración del par estéreo 1-2, P_2 son los puntos que devuelve la cámara 2 respecto a su propio sistema de referencia, y $P_{2|1}$ es el resultado final que se busca, los puntos de la cámara 2 con respecto al sistema de referencia de la cámara 1:

5. Calibración estéreo de cámaras ToF

$$P_{2|1} = R_{12}(P_2 - T_{12}) \quad (63)$$

Para referenciar la cámara 3, en primer lugar se utiliza la calibración del par estéreo 2-3 para referenciar la cámara 3 al sistema de la cámara 2 según la expresión (64):

$$P_{3|2} = R_{23}(P_3 - T_{23}) \quad (64)$$

Y ahora hay que referenciar $P_{3|2}$, que está referido a la cámara 2, a la cámara 1. Para ello se hace uso de la expresión (63):

$$P_{3|1} = R_{12}(P_{3|2} - T_{12}) = R_{12}(R_{23}(P_3 - T_{23}) - T_{12}) = \quad (65)$$

Por último, falta expresar los puntos de la cámara 4 con respecto a la cámara 1. Podría realizarse pasando primero al sistema de referencia de la cámara 3 mediante la calibración del par 3-4, y a partir de ahí aplicar la ecuación (65), y el resultado sería el de la expresión (66):

$$\begin{aligned} P_{4|3} &= R_{41}(P_4 - T_{41}) \\ P_{4|2} &= R_{32}(P_{4|3} - T_{32}) = R_{32}(R_{41}(P_4 - T_{41}) - T_{32}) \\ P_{4|1} &= R_{12}(P_{4|2} - T_{12}) = R_{12}(R_{32}(R_{41}(P_4 - T_{41}) - T_{32}) - T_{12}) \end{aligned} \quad (66)$$

Sin embargo, en cada paso se va acumulando el error de las 3 calibraciones utilizadas para los cambios de referencia, por lo que el resultado final acumulará más error. Para disminuir dicho error sería más conveniente utilizar los datos de la calibración del par estéreo 4-1 y directamente obtener los datos de la cámara 4 referenciadas a la primera por la siguiente operación más sencilla:

$$P_{4|1} = R_{41}P_4 + T_{41} \quad (67)$$

Con todo ello, se representan los sistemas de referencia de las cuatro cámaras con respecto al sistema de coordenadas de la cámara 1, y se obtienen las gráficas mostradas en la Figura 85:

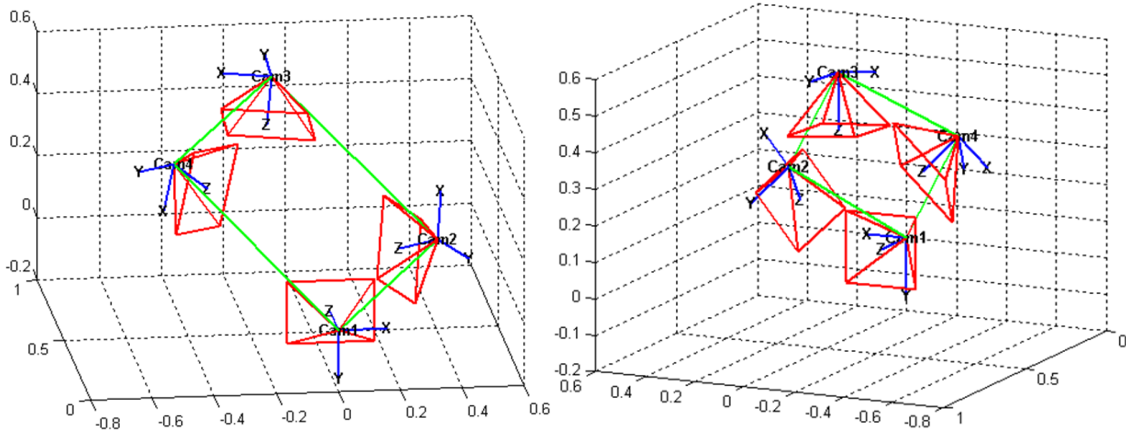


Figura 85. Sistema completo a partir de las calibraciones estéreo

A modo de comprobación, los orígenes de las cámaras deben coincidir con la posición en la que están colocadas en el sistema real si se considera la cámara 1 como origen. Por tanto, el cuadrilátero formado por dichos puntos (representado con líneas de color verde en la Figura

85) deberá tener unas medidas que coincidan con la distancia existente entre las cámaras en el mundo real. En la tabla de la Figura 86 se muestran algunas de estas medidas.

Cámaras	Real	Calculada	Error absoluto	Error relativo
1-2	44.6	44.81	0.21	0.47%
2-3	72.8	73.19	-0.39	0.53%
3-4	44.8	44.9	0.1	0.22%
4-1	72.4	72.6	0.2	0.27%
1-3	84.8	84.76	-0.04	0.04%
2-4	83.2	83.38	0.18	0.21%

Figura 86. Distancias reales y calculadas entre las cámaras del sistema

De nuevo, el error existente es bastante bajo, por lo que finalmente se concluye que la calibración de todo el sistema está correctamente realizada, pudiendo reconstruir puntos obtenidos mediante sus imágenes con un error menor al 2% en todos los casos.

6. Reconstrucción 3D

Una vez que se ha demostrado que todo el sistema está correctamente calibrado según los datos obtenidos a partir de las pseudo-imágenes en escala de grises, deben tomarse las superficies 3D obtenidas a partir de las cámaras de tiempo de vuelo a partir de su distancia radial y los parámetros de calibración intrínsecos y de distorsión (apartado 2.4), y aplicarles las rotaciones y traslaciones calculadas mediante las calibraciones estéreo, de manera que se logre que coincidan todas las nubes de puntos, con el objetivo de formar una única superficie con los datos de las 4 cámaras.

6.1. Reconstrucción 3D de una sola cámara

En primer lugar, cada una de las cámaras devuelve una pseudo-imagen de escala de grises, que va a ser llamada imagen de amplitudes, y una imagen de distancias. Mediante los datos de calibración del modelo pinhole y los parámetros de distorsión, junto con los datos de la imagen de distancia, se puede realizar una reconstrucción 3D desde el punto de vista de esa cámara. Un ejemplo de esto sería la segunda fila de la Figura 87. En estas imágenes, la escala de color representa las distancias de los puntos 3D a la cámara, siendo los colores fríos (azul) los más cercanos al sensor, y los calientes (rojo), los más lejanos. Para realizar esta escala de color es necesario ajustarla entre las distancias mínima y máxima de medida, es decir, que la escala de representación coincida con la de medida, que en la escena a reconstruir varía entre 0.15 cm y 1.10 m.

Por último, esta reconstrucción 3D puede ser mejorada de manera visual si en lugar de representar una escala de distancias se representan los datos de la imagen de amplitud asociados a cada uno de los puntos 3D. Con ello, se obtiene la fila inferior de la Figura 87, en la que efectivamente se observa cómo el color de la escena ha sido correctamente fusionado con los datos presentando, por ejemplo, el cubo un tono de gris claro y el fondo de la escena un tono de un color más oscuro.

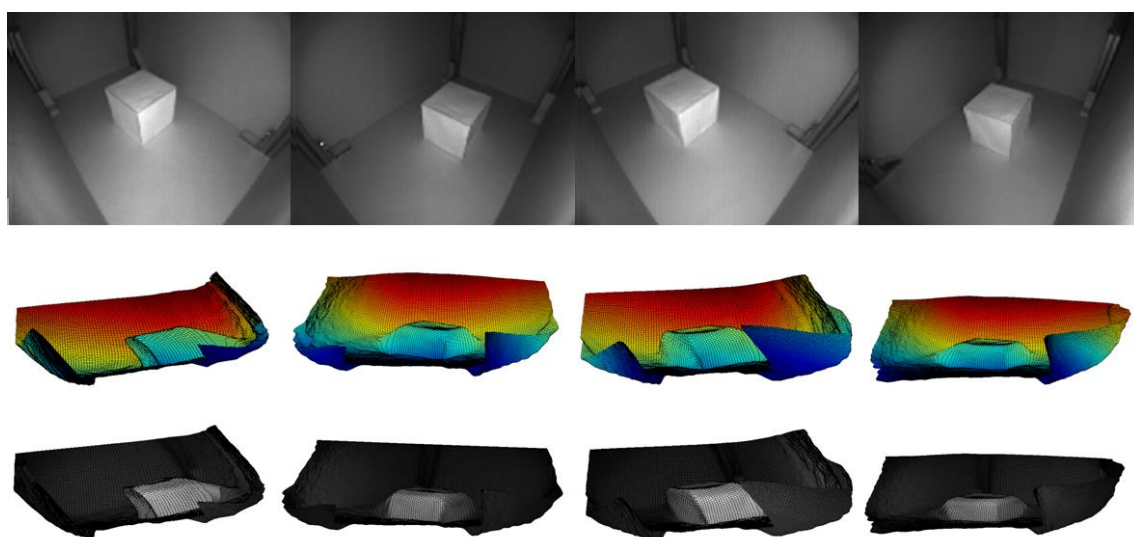


Figura 87. Reconstrucción 3D individual de un cubo por cada cámara. De izquierda a derecha, cámaras 1, 2, 3 y 4. De arriba abajo, imagen de amplitudes, reconstrucción 3D con escala de distancias y reconstrucción 3D con amplitudes.

En la Figura 88 se muestra otro ejemplo en el que se trata de reconstruir otra escena más compleja en la que aparecen dos poliedros: una pirámide de 3 caras y un polígono hexagonal. Esta vez la vista de la escena desde cada una de las cámaras varía en mayor medida que en el caso anterior dependiendo de la perspectiva de la cámara, es decir, del punto en el que esté colocada.

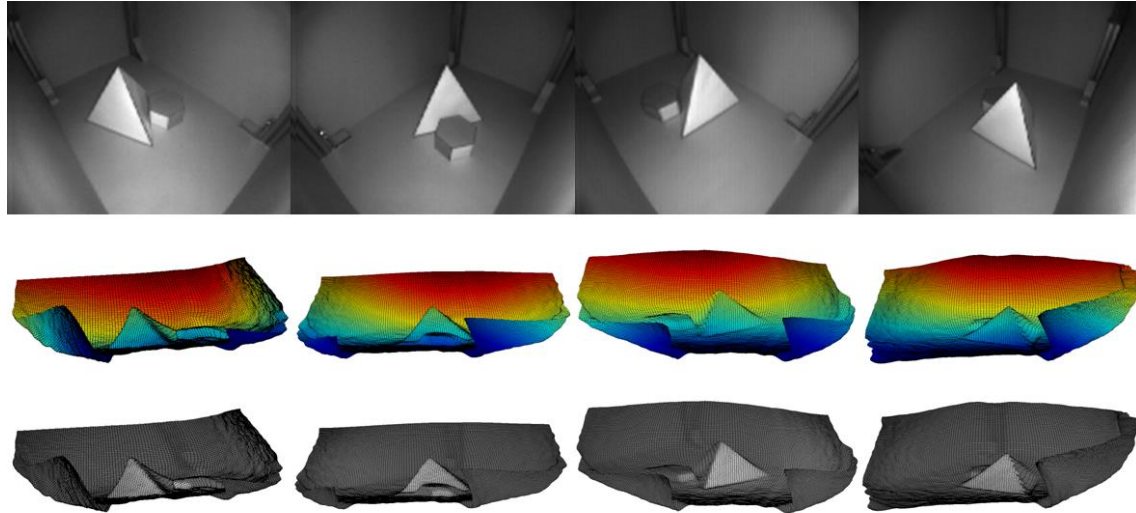


Figura 88. Reconstrucción 3D individual de una escena más compleja por cada cámara. De izquierda a derecha, cámaras 1, 2, 3 y 4. De arriba abajo, imagen de amplitudes, reconstrucción 3D con escala de distancias y reconstrucción 3D con amplitudes.

En estas figuras se aprecia que las cámaras capturan los elementos de la escena, pero la precisión o el detalle no son muy elevados. Por ejemplo, tiende a redondear las esquinas y los picos, así como las aristas del cubo no son perpendiculares al plano del suelo, ni las paredes de la escena forman ángulos de 90° entre sí y con el suelo. Esto es debido al efecto del multicamino, que precisamente se manifiesta de esta manera. Además, se debe tener en cuenta que la resolución es baja, por lo que no se tiene información con detalle de todos los puntos de la escena. En el apartado 6.1.2 se profundiza este aspecto para llegar a una conclusión sobre la precisión de la reconstrucción de una sola cámara, ya que esto afectará totalmente a la hora de fusionar los datos de más de una de ellas.

6.1.1. Eliminación de paredes

Debido a que las paredes del espacio construido (en el que se colocan los objetos a reconstruir en 3D) en todas las imágenes van a ser un elemento común a todas ellas, se ha decidido facilitar la opción de eliminar la mayor parte posible de las paredes para visualizar solamente en las gráficas aquellos objetos que se pretenden reconstruir, es decir, lo que realmente constituye información importante del escenario. Debido a que la escena es estática y las cámaras no se van a mover de su posición, para llevar a cabo una aproximación de eliminación de puntos de las paredes, podría bastar solamente con eliminar los píxeles de las zonas exteriores de la imagen (laterales y parte trasera). Los resultados obtenidos para las imágenes mostradas en la Figura 87 y la Figura 88 son los que aparecen en la Figura 89 y la Figura 90, respectivamente.

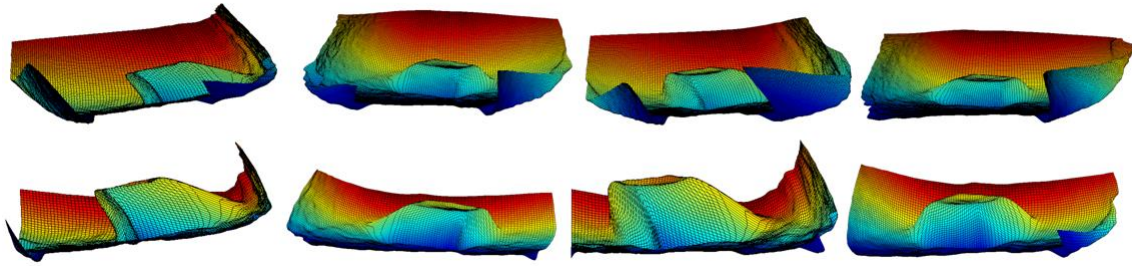


Figura 89. Superficie 3D del cubo con las paredes eliminadas

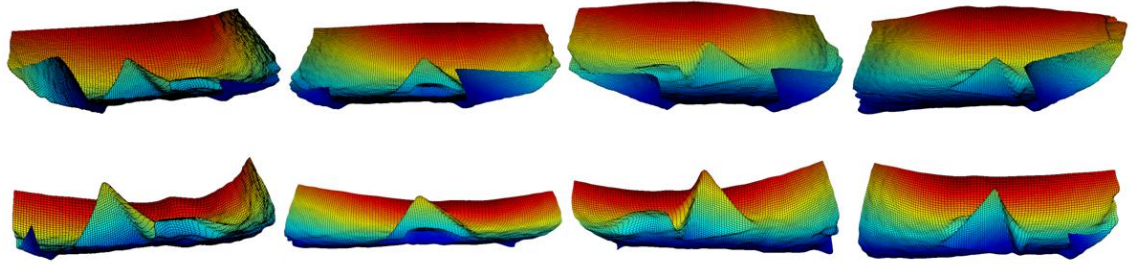


Figura 90. Superficie 3D de la escena más compleja con las paredes eliminadas

Como se aprecia, en las nuevas superficies 3D se visualizan mejor las zonas realmente importantes para la reconstrucción, que son los objetos posicionados sobre la escena. Y no solamente es el hecho de visualizar la zona de los objetos, sino que además la escala de color para representar la distancias se re-escala al nuevo rango de representación, por lo que las superficies de los puntos de los objetos aparecen representadas en un rango más amplio que cuando se tenían también en cuenta las paredes.

Además, a la superficie tridimensional resultante de esta eliminación o minimización en la escena de la aparición de paredes, también es posible adjuntarle la información de la imagen de amplitudes y representarla junto con su pseudo-escala de grises, como se muestra en la Figura 91 y la Figura 92

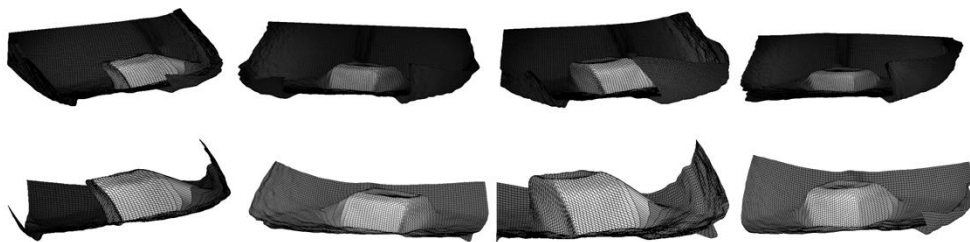


Figura 91. Superficie 3D del cubo con las paredes eliminadas y la pseudo escala de grises añadida



Figura 92. Superficie 3D de la escena más compleja con las paredes eliminadas y la pseudo escala de grises añadida

6.1.2. Análisis de precisión

De la misma manera que se concluyó que la calibración individual de las cámaras y la estéreo de cada par eran suficientemente fiables como para realizar una reconstrucción con un error menor al 2% en todos los casos utilizando las imágenes de escala de grises, es importante conocer la precisión de los datos representados mediante las superficies tridimensionales obtenidas a partir de las imágenes de distancia. Esto es importante porque si el error es alto, las superficies calculadas no se corresponderán; esto quiere decir que al aplicar las matrices de rotación y traslación a dos nubes de puntos de cámaras distintas para hacerlas coincidir y fusionar sus datos, no se va a lograr que dichos datos se superpongan, y el error de fusión de datos será alto.

Las coordenadas bidimensionales de los píxeles de la pseudo-imagen de escala de grises y de la imagen de distancias que proporciona las cámaras coinciden. Es decir, si un punto 3D de la escena (por ejemplo, el vértice de una pirámide) se representa en la imagen de amplitudes en las coordenadas (x,y) , las coordenadas de la imagen de distancias (x,y) se corresponden con la distancia a la que está de la cámara ese mismo punto 3D. Esto quiere decir que en este sentido son aplicables también los mismos parámetros de distorsión de la cámara en ambas imágenes.

Lo que en realidad hay que comprobar es si la distancia medida con las cámaras se corresponde con la distancia real existente. Para ello se va a capturar una imagen de distancias, como la que aparece en la Figura 93 a la derecha, y se va a medir el valor de distancia medido que devuelve la cámara de algunos de los píxeles. Por otro lado, conociendo las medidas métricas reales aproximadas de dichos puntos característicos en la escena, se podrá llevar a cabo una comparación que establezca la precisión del proceso. Se ha decidido utilizar para esta comparación una caja rectangular por tener bastantes puntos representativos: sus esquinas.

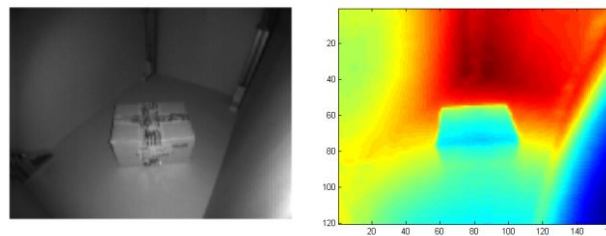


Figura 93. Imagen de pseudo-escala de grises (izquierda) y de distancias (derecha).

Para comenzar con la comparación, se van a medir las distancias reales a cada una de las esquinas visibles de la caja rectangular con las distancias que resultan en la imagen de profundidad. Los puntos que van a ser medidos se muestran en la imagen de amplitudes en la Figura 94, y se comparan en la tabla de la Figura 95.



Figura 94. Puntos de los cuales se está midiendo la distancia

Punto	Distancia Real aprox[cm]	Calculada[cm]	Error [cm]
1	46	54.8	8.8
2	48.5	57	8.5
3	60	70.5	10.5
4	61	72	11
5	57.5	67.8	10.3
6	58	68.8	10.8

Figura 95. Comparación de distancias

Obviamente, los resultados de distancia que proporciona la cámara de tiempo de vuelo no son aceptables. Estos son los datos que directamente proporciona el sensor de tiempo de vuelo, con la corrección de la distorsión según los parámetros de calibración, que se demostró que es correcta. Por lo tanto, si se calcula la superficie tridimensional a partir de estos datos (Figura 96) de distancia que contienen en torno al 18% de error en la medida, la superficie 3D incorporará este error en sus datos.

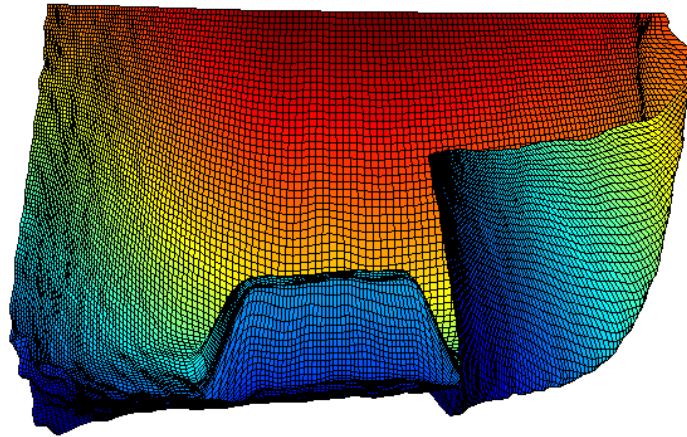


Figura 96. Reconstrucción 3D de una caja rectangular situada en la escena.

Es decir, la distancia que las cámaras proporcionan de manera intrínseca a su funcionamiento incorpora un error muy alto y que no es para nada aceptable. La solución a este error sería realizar otro proceso de calibración para ajustar las distancias medidas a las distancias reales. Sin embargo, este objetivo no entra dentro del presente Trabajo de Fin de Máster. Por lo tanto, debido a que cada cámara proporciona medidas con un alto error, como se acaba de demostrar, al tratar de fusionar datos, los errores de cada cámara se sumarían, haciendo imposible una reconstrucción 3D precisa

No obstante, se observa que a priori la cámara está proporcionando los datos con un determinado error porcentual que podría ser corregido mediante la aplicación de un factor de escala. Este paso no soluciona el error expuesto, ya que éste no es constante en todas las zonas de la imagen y varía en función de la propia distancia, el color y el tipo de material. Sin embargo, se va a tratar de corregir dicho factor de escala en la medida de lo posible en la imagen de distancias con el objetivo de guiar los pasos de una reconstrucción 3D aproximada. Con la corrección del factor de escala se vuelven a capturar imágenes y se obtiene la siguiente tabla:

Punto	Distancia Real [cm]	Calculada [cm]	Error absoluto [cm]
1	46	47.44	1.44
2	48.5	50.23	1.73
3	60	60.38	0.38
4	61	60.87	-0.13
5	57.5	56.45	-1.05
6	58	57.15	-0.85

Figura 97. Comprobación de distancias tras la corrección del factor multiplicativo

Como se observa, tras la corrección del factor de escala de la distancia ahora las distancias calculadas mediante el sensor ToF toman un valor con un error más bajo y para algunas aplicaciones, aceptable. Por lo tanto, las medidas tomadas directamente con las cámaras, como las del apartado 6.1.1, guardan una relación de proporción de escalas orientativa, pero no tienen medidas correctas. Si se quiere que los datos se ajusten a la realidad, y de esta manera poder hacer que coincidan las nubes de puntos de todas las cámaras mediante los datos de calibración estéreo calculados, sería necesario corregir el factor de escala, es decir, calibrar distancias. Posiblemente este factor de escala de profundidad debería incluirse en un modelo de calibración global de la cámara para optimizar todos los parámetros simultáneamente. Pero para mostrar algunos ejemplos de reconstrucción de manera orientativa, se va a utilizar este factor de corrección medio.

6.2. Fusión de datos de dos cámaras

En este apartado se van a fusionar las nubes de puntos de 2 cámaras como paso inicial a la reconstrucción 3D total, con los datos de todas las cámaras del sistema. De esta manera se podrá comprobar si la corrección del factor de escala de distancia es efectiva, así como si los datos estéreo de calibración (rotación y traslación) pueden ser aplicados satisfactoriamente también a los datos tridimensionales.

Para ello, se va a colocar de nuevo el cubo en la escena, se van a adquirir imágenes con las cámaras 1 y 2, se corrige el factor de escala y se van a ir exponiendo los resultados obtenidos a continuación. En primer lugar, se van a mostrar las imágenes de amplitudes desde ambas cámaras para saber cómo es la escena que se pretende reconstruir.

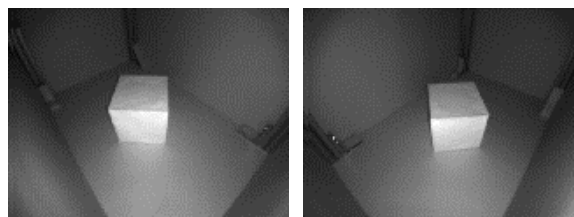


Figura 98. Imagen de la escena a reconstruir desde la cámara 1(izquierda) y desde la cámara 2 (derecha)

A continuación, en la Figura 99 se muestran las superficies tridimensionales obtenidas para cada una de las cámaras, para cada una de las vistas anteriores, junto con el sistema de referencia de su cámara representado.

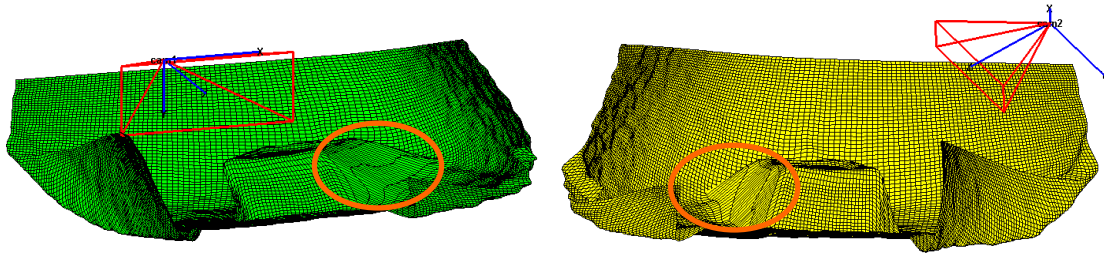


Figura 99. Superficie 3D de la cámara 1 (izquierda) y de la cámara 2 (derecha). Problema superficies ocultas.

Cuando se utiliza la función *surf* de MATLAB para representar superficies, una todos los puntos 3D obtenidos formando una única malla. Esto quiere decir que en las zonas en las que no existan puntos porque no sean visibles desde la perspectiva de la cámara, en lugar de mostrar esa zona sin representación porque no se tiene información, una los dos puntos más cercanos de los que se tenga información. Esto es lo que ocurre en las caras ocultas del cubo para cada cámara. Se ha marcado este efecto con dos circunferencias naranjas en la Figura 99. Si se quiere eliminar este efecto visual, se pueden representar directamente los puntos tridimensionales, sin formar una superficie unida que los represente, y de esta manera no habrá puntos en las zonas en las que no existan datos, como se muestra en la Figura 100. Esto ayudará a la hora de superponer dos o más vistas estereo, ya que si no, la superficie generada en las zonas ocultas puede tapar a la otra posible superficie real de la otra representación cuya zona sea visible desde esa cámara.

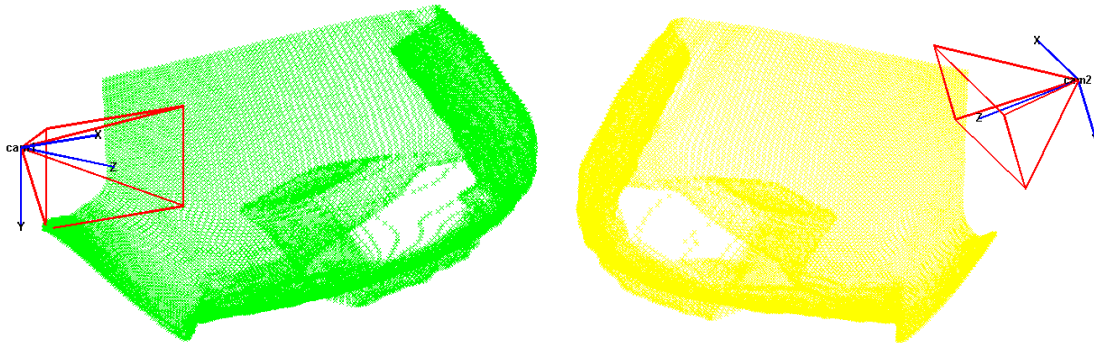


Figura 100. Representación 3D de la nube de puntos de la cámara 1 (izquierda) y la cámara 2 (derecha)

El siguiente paso es aplicar las matrices de rotación y traslación a la nube de puntos de la cámara 2 para que coincida con la nube de puntos de la cámara 1, tal y como se muestra en la Figura 101 y según la ecuación (68):

$$P_{2|1} = R_{12}P_2 + T_{12} \quad (68)$$

Como se aprecia, se ha conseguido rotar y trasladar la nube de puntos de manera ‘correcta’ a priori y el cubo coincide en ambas superficies (observar como coinciden las aristas y las esquinas en la izquierda de la Figura 101).

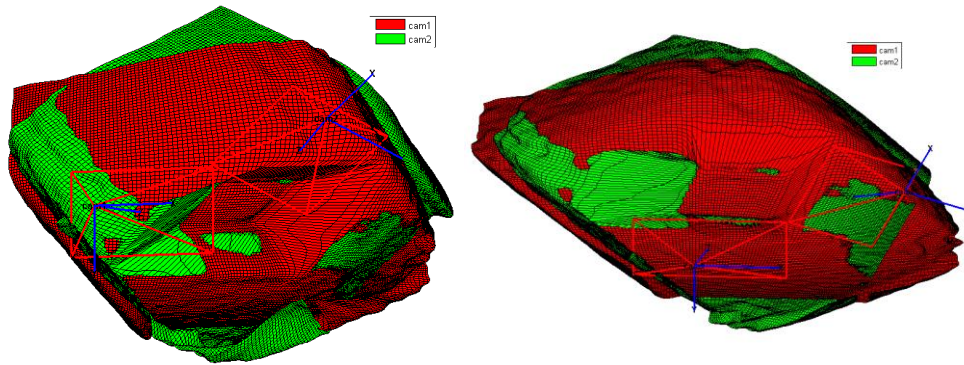


Figura 101. Superficies de la cámara 1 y 2 referidas al mismo sistema de coordenadas mediante aplicación de los parámetros del sistema estéreo.

Sin embargo, el problema en la representación en las zonas ocultas hace que se interpole información y se solape información real de la otra vista. Por lo tanto, a continuación se muestran los mismos datos representados en formato de nube de puntos, en lugar de como superficie tridimensional, y se puede observar en la Figura 102 como las zonas en las que no existen datos de la superficie no son interpoladas, si no que permanecen sin puntos.

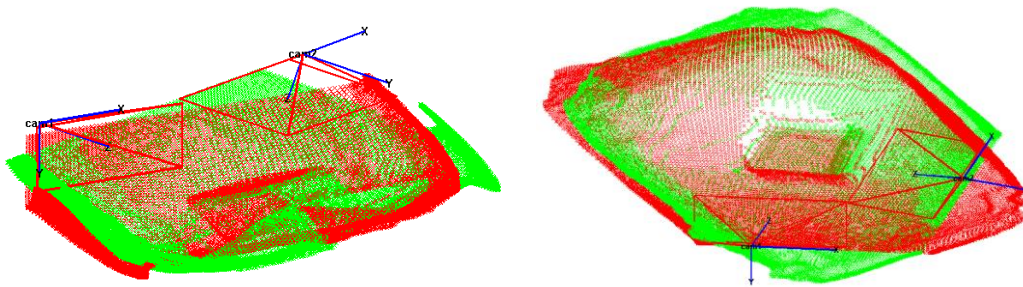


Figura 102. Nubes de puntos de la cámara 1 y 2 referidas al mismo sistema de coordenadas mediante aplicación de los parámetros del sistema estéreo.

De todas formas, cuando se realiza una reconstrucción 3D se busca como objetivo mostrar una única superficie o nube de puntos que contenga la información de ambas vistas. Para ello, se va a fusionar los datos aplicando una búsqueda del punto más cercano de la nube de puntos de la cámara 2 (rotada y trasladada) con la cámara 1. Esta búsqueda se realiza mediante el algoritmo de k-vecinos más cercanos [22] que proporciona MATLAB®. De esta manera, se establecerá una correspondencia 2 a 2 entre los puntos de ambas nubes, y se realizará una media aritmética entre ellos para obtener una única superficie final promediada. En la Figura 103 se muestra el resultado de esta fusión de datos representado en formato superficie.

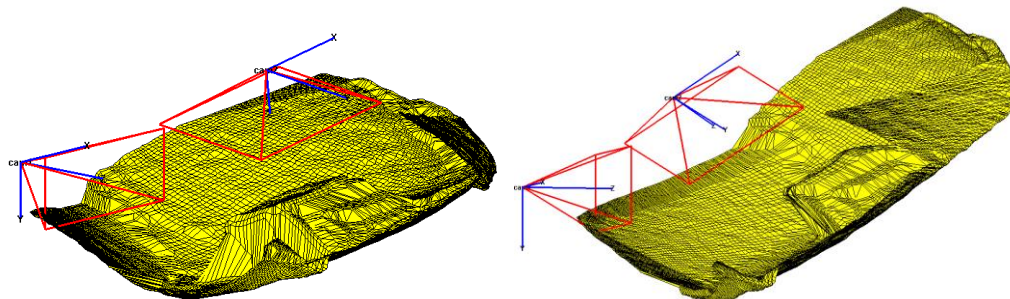


Figura 103. Fusión de datos de las dos nubes estéreo.

Se va a realizar otra representación con la escala de distancias para obtener una vista más atractiva de la imagen resultante en la Figura 104.

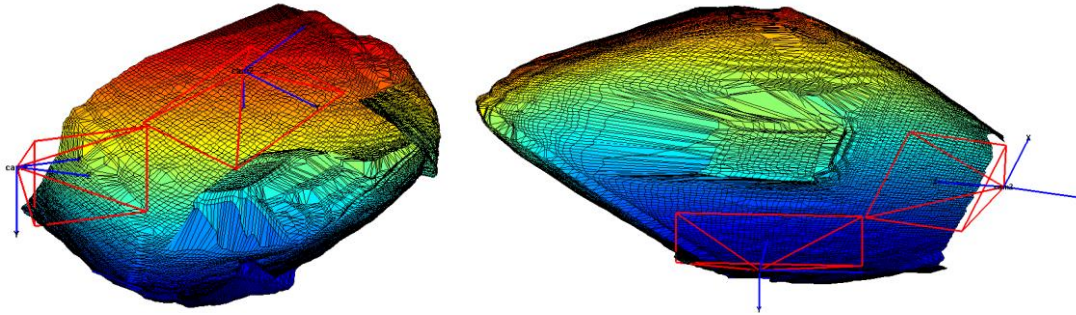


Figura 104. Fusión de datos de las dos nubes estéreo con escala de distancias

Como se observa, se ha conseguido reconstruir la caja desde la perspectiva de las dos cámaras utilizadas. Sin embargo, aunque a priori parece ser que la reconstrucción ha sido correcta, se deben medir las distancias de los puntos característicos del cubo (las cuatro esquinas de las aristas de la cara superior, de manera similar a las de la Figura 94) a la cámara 1 en la realidad y comprobar si se aproximan a la de la reconstrucción 3D. Se muestran las magnitudes de estas medidas en la tabla de la Figura 105.

Punto	Distancia Real [cm]	Calculada [cm]	Error absoluto
1	44.5	48.19	3.69
2	58.5	57.9	0.6
3	59	61.3	2.3
4	47	50	3
1-2	17	13	4
2-3	17	13.5	4.5

Figura 105. Tabla de errores de distancia en la reconstrucción 3D estéreo.

Partiendo de unas nubes de puntos obtenidas, y teniendo en cuenta que la nube de puntos final es una media de los puntos de ambas, era predecible que el error final aumentase, llegando a tener en estos puntos hasta un error de más de 4 cm en un rango de medida menor de 60 cm, lo que obviamente es mejorable para reconstrucción precisa de escenas. Esto indica que el factor de corrección no es simplemente un coeficiente lineal por el que multiplicar los datos de distancia, sino que se debe realizar otra calibración de este parámetro para conseguir datos finales más fiables.

A continuación, se va a realizar otro ejemplo de reconstrucción 3D mediante otro par de cámaras, se va a reconstruir la pirámide de tres caras laterales con la que se trabajó en el apartado 6.1, esta vez mediante las cámaras 3 y 4 y sus correspondientes matrices de traslación y rotación. Tal y como se ha concluido, las medidas no son muy precisas, pero se consigue una reconstrucción orientativa de la escena. En primer lugar, en la Figura 106 se muestran las imágenes de amplitudes tomadas por ambas cámaras.

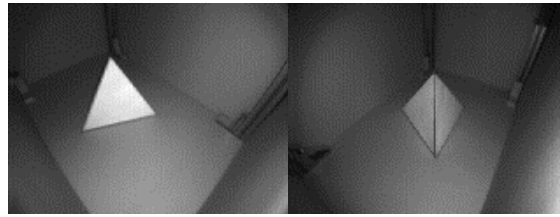


Figura 106. Imagen de la escena a reconstruir desde la cámara 3 (izquierda) y desde la cámara 4 (derecha)

A continuación se expondrán los resultados, análogos al caso del cubo con las cámaras 1 y 2. En primer lugar, las superficies obtenidas desde cada una de las cámaras se muestran en la Figura 107:

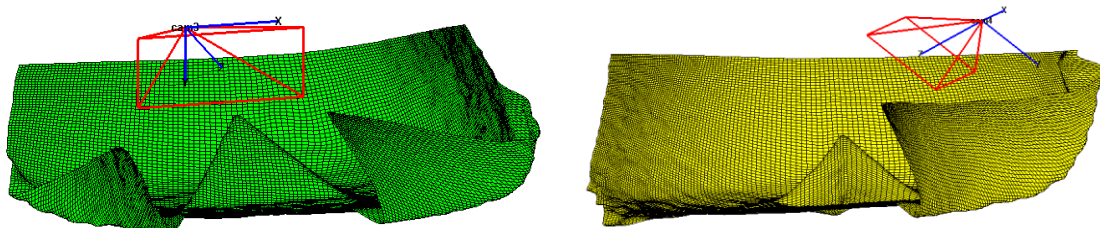


Figura 107. Superficie 3D de la cámara 3 (izquierda) y de la cámara 4 (derecha).

En la Figura 108 se muestran ambas superficies referenciadas con respecto al sistema de la cámara 3, tras aplicar la traslación y la rotación a los puntos 3D de la cámara 4, aplicando los parámetros de la calibración estéreo de dichas cámaras. Como se observa, el pico de la pirámide no coincide en ambas superficies, sino que está desplazado 6 cm entre sí. Esto es previsible debido a los errores que se han explicado y se van acumulando, y también hay que tener en cuenta que es posible que la calibración estéreo de las cámaras haya variado ligeramente por algún movimiento mínimo de alguna de ellas, y las matrices de rotación y/o traslación se hayan visto levemente modificadas desde el momento en el que se calibraron.

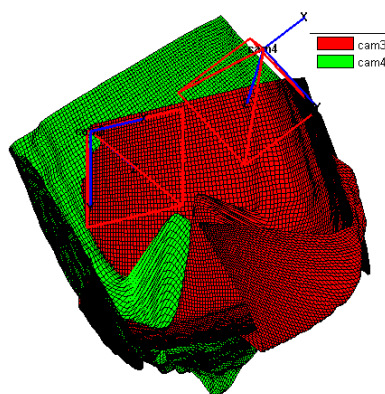


Figura 108. Superficies de la cámara 3 y 4 referidas al mismo sistema de coordenadas mediante aplicación de los parámetros del sistema estéreo.

Si se fusionan las superficies de ambas pirámides según el método de búsqueda de k-vecinos más próximos y media aritmética, se obtiene una superficie como la mostrada en la Figura 109 desde dos puntos de vista distintos, en la que al realizarse la media de los puntos, se

consigue finalmente una pirámide que está ligeramente deformada (se observa mejor en la figura de la derecha).

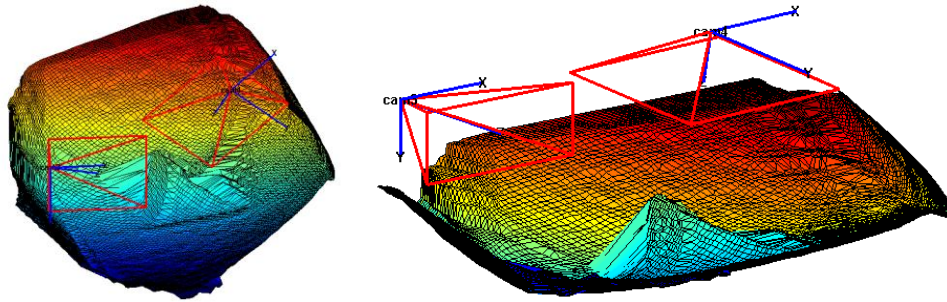


Figura 109. Fusión de datos de las dos nubes estéreo

Si se quiere mejorar esta deformación, ya que no se va a conseguir una reconstrucción totalmente fiable, pero se puede buscar una representación orientativa, se puede aplicar un algoritmo iterativo ICP-GPA que toma ambas nubes de puntos y halla la rotación y traslación que debe aplicarse a una de ellas para lograr que ambas coincidan en la mayor medida posible. Tras la aplicación de este algoritmo, el par estéreo se ve representado en la Figura 110.

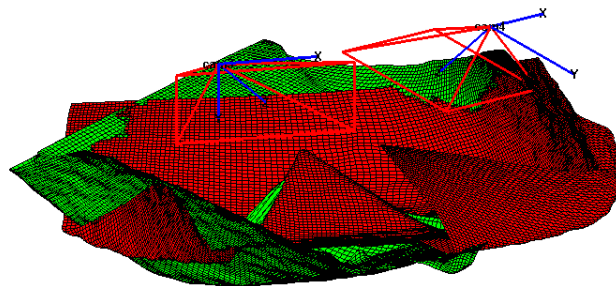


Figura 110. Nubes de puntos alineadas tras ejecutar Iterative Closest Points

Por último, a las nuevas nubes de puntos 3D alineadas mediante el algoritmo ICP se le aplica la fusión de datos, obteniendo finalmente la reconstrucción 3D de la pirámide como se muestra en la Figura 111.

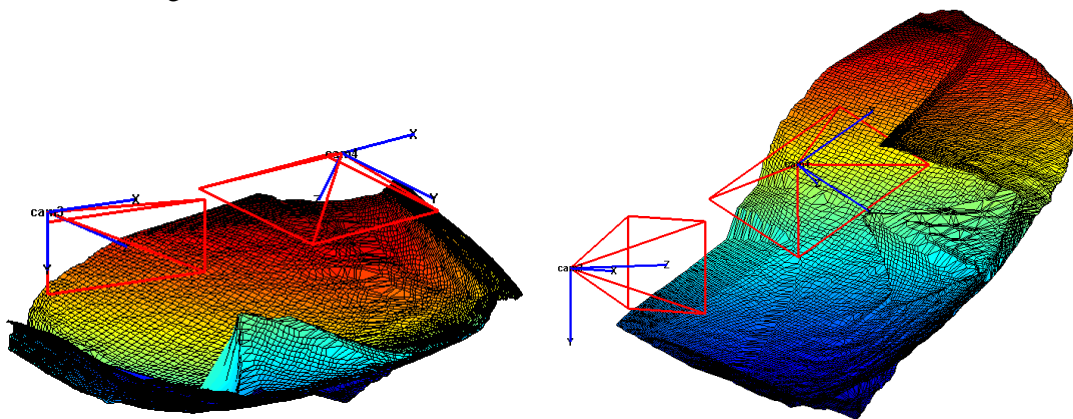


Figura 111. Reconstrucción 3D final de la pirámide con las cámaras 3 y 4

Si se calculan las coordenadas del pico de la pirámide, así como del corte de la arista común a ambas cámaras para calcular el error cometido en este caso, y se halla su distancia al origen de referencia de la cámara 3, tanto en el mundo real como en la representación 3D se obtienen las siguientes comparaciones:

Punto	Distancia Real [cm]	Calculada [cm]	Error absoluto [cm]
Pico	50	56.7	6.7
Corte arista-suelo	63	65.16	2.16
Medida arista	28.5	23.72	4.78

Figura 112. Tabla de comparación entre distancias en la reconstrucción 3D y medidas reales

Se observa que los errores de la reconstrucción siguen siendo elevados, a pesar de que este algoritmo iterativo presenta la ventaja de unificar lo máximo posible las superficies para obtener una malla final representativa de la escena, aunque sus medidas sean erróneas. Además, este algoritmo podría presentar una ventaja muy significativa para un futuro paso en el que se calibraran las distancias y se obtuviera una representación más precisa. A pesar de que las cámaras están colocadas en un punto fijo, cualquier desplazamiento que pueda ocurrir en su posición hace que la situación relativa entre ésta y sus dos cámaras contiguas cambie, y por tanto, los parámetros de rotación y traslación que fueron calculados mediante las calibraciones estéreo también lo harán, y ya no son válidos. Por lo tanto, el algoritmo iterativo ICP podría conseguir que se puedan compensar estos pequeños movimientos sin necesidad de volver a calcular la calibración estéreo.

Como conclusión a este apartado se puede indicar que no se logra reconstruir la escena en 3D con un error reducido, que se sabe que este error proviene de que la medida de distancia es errónea porque debe ser calibrada, pero que sin embargo, es posible realizar una reconstrucción 3D aproximada de la escena. Si se quiere alcanzar una reconstrucción precisa con estas mismas cámaras sería necesario realizar, aparte de la calibración con las imágenes de pseudo escala de grises, una calibración de distancias para corregir correctamente el factor de escala y un posible offset.

6.3. Reconstrucción 3D con las cuatro cámaras

Por último, se van a aunar los resultados de las 4 cámaras para poder representar las 4 nubes de puntos como una sola, consiguiendo una superficie final orientativa del objeto, es decir, su reconstrucción tridimensional. Se va a reconstruir el cubo en primer lugar para realizar la reconstrucción y analizar sus resultados. Si en la reconstrucción con 2 cámaras el error que se acumulaba era tal que se debía considerar, al tener que fundir los datos de más cámaras, se espera que el error sea mayor. Se muestran los resultados para poder observar la eficacia en la representación orientativa de la escena. En la Figura 113 se muestra la imagen de pseudo escala de grises obtenida desde cada una de las cámaras, y en la Figura 114 aparece representada la superficie 3D que se observa de cada una de las 4 cámaras.

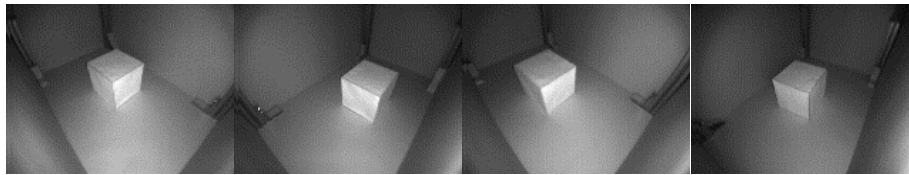


Figura 113. De izquierda a derecha, imágenes de amplitud de la escena a reconstruir desde la cámara 1 a la 4.

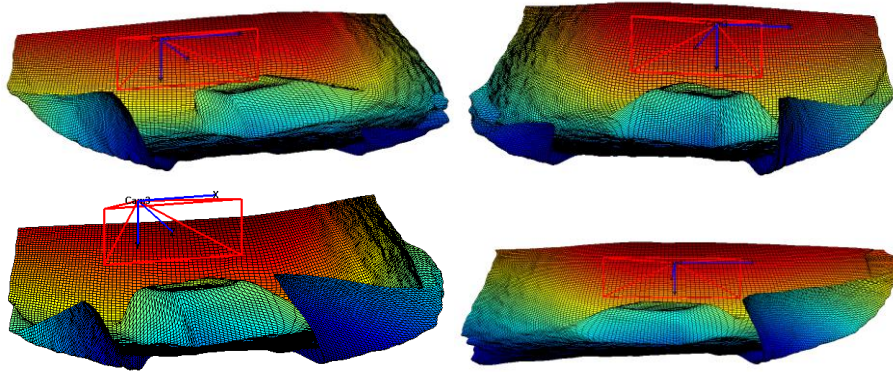


Figura 114. De izquierda a derecha y de arriba abajo, superficies 3D de las cámaras 1,2,3 y 4.

A continuación se muestran los datos de las 4 superficies rotadas y trasladadas en la Figura 115, cada una de ellas representadas mediante un color, como se muestra en la leyenda, y para visualizar el cubo de manera más sencilla, se eliminan las paredes en la Figura 116 .

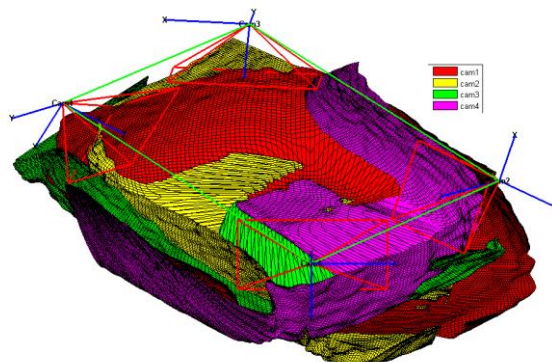


Figura 115. Superficies de las cuatro cámaras referidas al mismo sistema de coordenadas mediante aplicación de los parámetros de los cuatro pares estéreo.

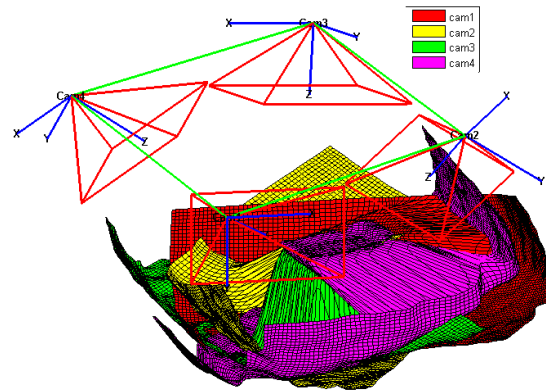


Figura 116. Figura 115 con las zonas de las paredes minimizadas

Por último, se fusionan todas las superficies con el objetivo de formar una única malla para describir el cubo y se obtienen los resultados de la Figura 117.

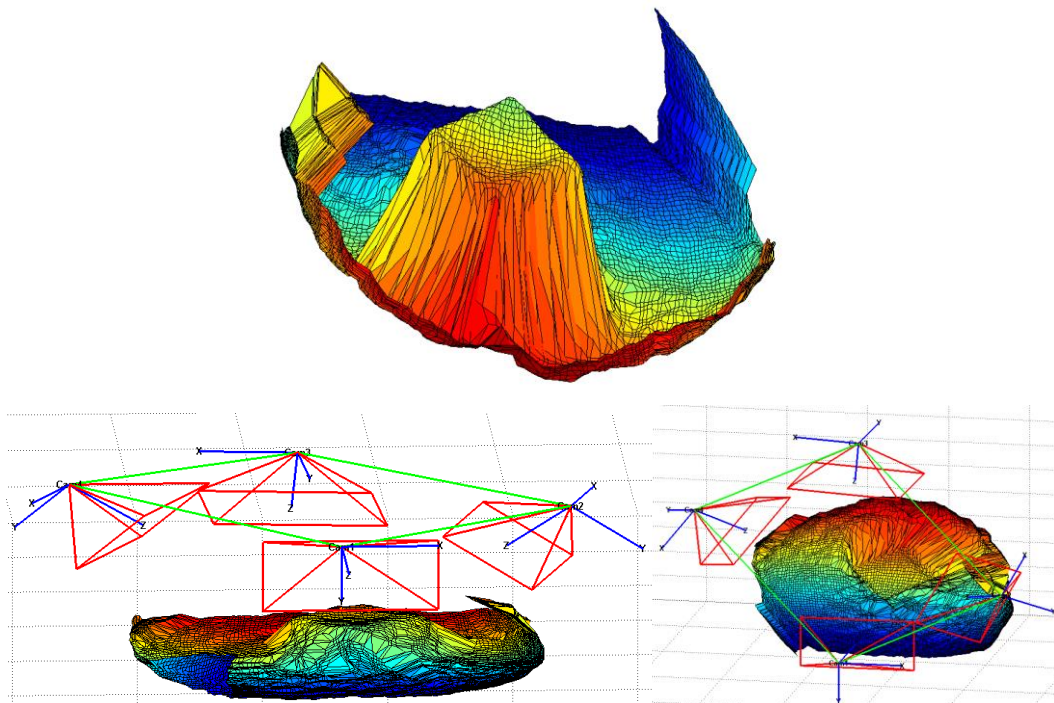


Figura 117. Arriba: cubo reconstruido. Abajo: cubo reconstruido con respecto a las cámaras desde dos perspectivas distintas

El lado del cubo real es de 17 cm, y si se miden los lados obtenidos en superficie de arriba de la Figura 117 se tiene un lado de 13 cm y otro de 11 cm, por lo que como primera conclusión se observa que el cubo deja de tener una forma cuadrada en su cara superior, y que además existe un error de un con respecto a las medidas reales, por lo que de nuevo se corrobora que el error es alto para realizar una reconstrucción precisa, si bien se puede obtener una aproximación de ella.

7. Toolbox realizada

Con el objetivo de hacer más accesible todo el proceso explicado a lo largo de la memoria, se ha desarrollado una sencilla interfaz para guiar al usuario a través de todo el método de calibración, reconstrucción en tres dimensiones y visualización gráfica de los resultados obtenidos.

7.1. Organización del directorio de la interfaz

Por facilidad de manejo de la interfaz diseñada, se sugiere que se cree un directorio que contenga una serie de subcarpetas como las que se muestran en la Figura 118. No obstante, junto con el archivo de datos que se proporciona al usuario ya viene el directorio generado para poder ser utilizado sin necesidad de crear uno nuevo si no se desea.

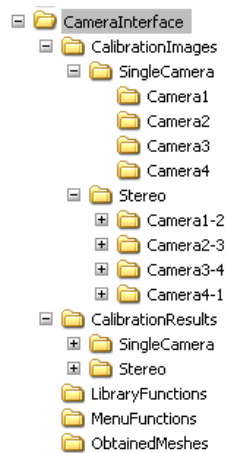


Figura 118. Directorio con carpetas y subcarpetas necesarias para el uso de la toolbox

- ‘*CalibrationImages*’: subcarpeta que contiene las pseudo-imágenes de escala de grises para realizar la calibración. Dentro de esta carpeta, hay dos divisiones:
 - ‘*SingleCamera*’: se almacenan las imágenes utilizadas para la calibración de las cámaras de manera individual. Cada cámara utilizada tendrá una propia carpeta para sus imágenes.
 - ‘*Stereo*’: se guardarán las imágenes utilizadas para la calibración estéreo de cada par de cámaras calibradas. Habrá una sub-carpeta para cada par de cámaras.
- ‘*CalibrationResults*’: en esta carpeta se van a almacenar los datos que resulten de calibraciones realizadas. En consecuencia, aquí es donde hay que dirigirse para cargar resultados guardados. También se organizarán en sub-carpetas, igual que la carpeta ‘*CalibrationImages*’.
- ‘*LibraryFunctions*’: en este directorio se almacenan todos los scripts de MATLAB® .m que realizan los procesos y algoritmos desarrollados a lo largo del trabajo. Del mismo modo, también hay que situar en esta carpeta los archivos necesarios para poder acceder a las cámaras de tiempo de vuelo conectadas al PC, los drivers y los ficheros .dat que contienen los parámetros de calibración de cada cámara.
- ‘*MenuFunctions*’: aquí se encuentran los scripts de MATLAB® necesarios para mostrar el menú interactivo diseñado para que el usuario navegue por las opciones disponibles simplemente haciendo click en las opciones deseadas.

- *'ObtainedMeshes'*: en esta carpeta se almacenan las nubes de puntos obtenidas por las cámaras conectadas al ordenador. Por lo tanto, aquí será donde el usuario tendrá también que dirigirse cuando deba seleccionar varias vistas de la escena para realizar una reconstrucción tridimensional.

En el directorio inicial, *'CameraInterface'*, se encuentra el script de MATLAB® *'InitMenu.m'* que al ejecutarse abre una figura con el menú desplegable inicial de la aplicación diseñada, mostrado en la Figura 119, desde el que se tiene acceso a todas las funcionalidades.

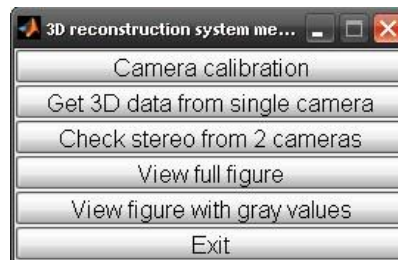


Figura 119. Menú principal de la interfaz.

Desde este menú se puede acceder a la parte de calibración de las cámaras, tanto para realizar una nueva calibración, como para cargar y comprobar datos previamente existentes, se permite también obtener nuevas imágenes directamente de las cámaras conectadas y su nube de puntos en 3D si la cámara está calibrada, así como visualizar gráficamente las nubes de puntos de una sola cámara, de una vista estéreo o del sistema de las cuatro cámaras simultáneamente. A continuación se explica independientemente cada una de estos módulos.

7.2. Calibración de las cámaras

En el menú de calibración, como se aprecia en la Figura 120, se permiten las opciones de cargar datos de calibraciones que se hayan almacenado previamente, realizar un nuevo proceso de calibración o comprobar la validez de una calibración realizada (de los parámetros intrínsecos de la cámara). Siempre se da la opción de volver al principio del menú haciendo click en la casilla *'Home'*.

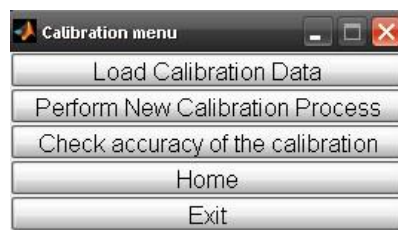


Figura 120. Menú de calibración

Si se desea cargar datos de calibración almacenados en memoria, se despliega otro menú (Figura 121) en el que pregunta si se quieren cargar datos de una calibración de una sola cámara o una calibración estéreo de un par de cámaras.



Figura 121. Menú de carga de calibración

Tanto si se elige cargar datos de una sola cámara o de un conjunto estéreo, se va a abrir una ventana (Figura 122) para seleccionar el directorio en el que se encuentran los archivos de calibración almacenados. Además, dependiendo de cuál de las dos opciones se elija, se abrirá por defecto la carpeta en la que en teoría deberían estar almacenados dichos datos, según se ha explicado en la distribución del directorio.

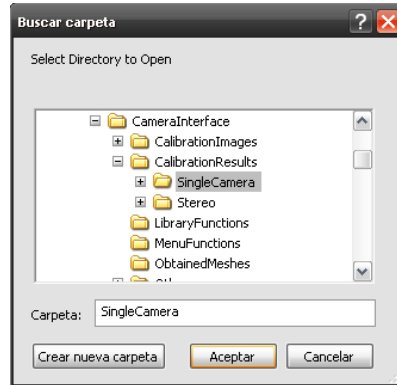


Figura 122. Selección del directorio para cargar los datos de calibración

Una vez que se ha seleccionado el directorio en el que está el archivo que contiene los datos de calibración, el programa está diseñado para cargar el archivo *.mat* que por defecto genera MATLAB® cuando se guarda una calibración. En caso de que sean los datos correspondientes a una calibración de una sola cámara, el nombre que tiene el archivo es *'Calib_Results.mat'*, y si es estéreo, *'Calib_Results_Stereo.mat'*.

Otra opción existente en el menú de calibración es la de comprobar la precisión de la calibración, basándose en las medidas de linealidad de la sección 4.3.2. Para ello, es necesario tener cargados unos datos de calibración y seleccionar también el directorio donde se encuentran las imágenes a eliminar la distorsión para realizar los cálculos.

La última opción disponible en este apartado de calibración es la de realizar una nueva calibración. Si se selecciona esta opción, en primer lugar se abre un menú como el de la Figura 121 en el que se pregunta el tipo de calibración que se quiere desarrollar.

7.2.1. Nueva calibración de una cámara

Cuando se desea calibrar una cámara conectada al PC, en primer lugar hay que tomar imágenes del patrón en la escena con la cámara, y almacenarlas en su correspondiente carpeta. A continuación, al seleccionar la opción de calibración se abre una ventana similar a la de la Figura 122 para elegir este directorio en el que se encuentran las imágenes.

A continuación, se muestra el nombre de los archivos existentes en el directorio, y se pedirá elegir las imágenes a utilizar para la calibración, como aparece en la Figura 123.

```

Basename camera calibration images (without number nor suffix): cam1_
Image format: ([]='r'='ras', 'b'='bmp', 't'='tif', 'p'='pgm', 'j'='jpeg', 'g'='jpeg', 'm'='ppm') b

Checking directory content for the calibration images (no global image loading in memory efficient mode)
Found images: 1...2...3...4...5...6...7...8...9...10...11...12...13...14...15...16...17...18...19...20...21...22
done

```

Figura 123. Selección de las imágenes de calibración

Entonces comienza el proceso de extracción de puntos característicos en cada una de las imágenes, como se explicó en la sección 3. Para cada imagen se irá abriendo secuencialmente una ventana como la de la Figura 124 para facilitar la búsqueda de los centroides.

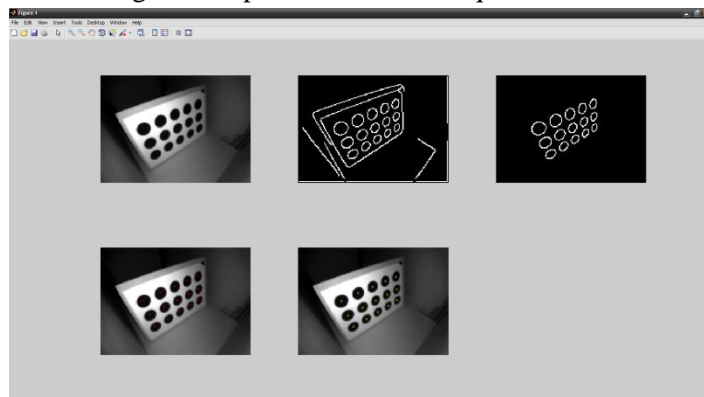


Figura 124. Extracción de puntos característicos en una imagen

Una vez que los centroides hayan sido correctamente hallados, se abre otra ventana en la que se pide al usuario que haga click sobre el centroide inicial para realizar la ordenación de centroides, como aparece en la Figura 125.

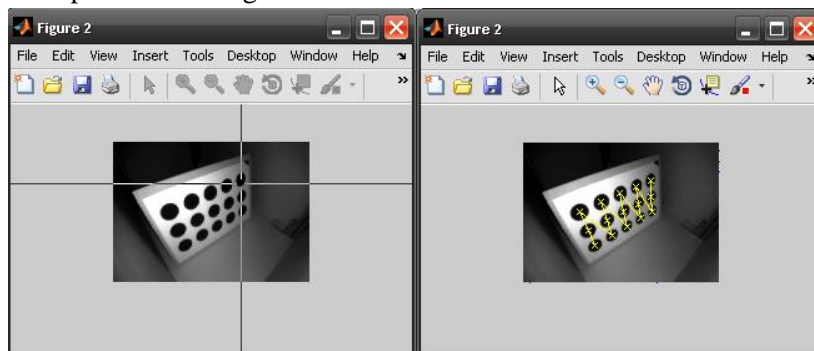


Figura 125. Click en centroide inicial (izquierda) y centroides ordenados (derecha)

Cuando se han obtenido los centroides de todas las imágenes, se ejecuta la función de calibración obtenida de la toolbox de Yves-Bouguet y la aplicación abre otra ventana emergente para preguntar por el directorio en el que se quiere guardar la calibración realizada.

7.2.2. Nueva calibración estéreo

Para realizar una calibración estéreo es necesario partir de los archivos de calibración de la cámara derecha y el de la izquierda. Estos archivos se obtienen según el proceso del apartado 7.2.1, y por comodidad se suelen llamar '*Calib_Results_Right.mat*' y '*Calib_Results_Left.mat*' respectivamente. Lo primero que pregunta la interfaz programada es por el directorio en el que se encuentran estos archivos. Para el proceso que se realiza en este trabajo no es necesario que las imágenes de la calibración estéreo se encuentren también directamente en este directorio.

Tras elegir los archivos de ambas cámaras, se utiliza la función de calibración de la toolbox de Yves-Bouguet estéreo, y devuelve los parámetros intrínsecos de cada una de las dos cámaras optimizados, y el vector de rotación y traslación calculado, como se muestra en la Figura 126. Por último, se elige el directorio donde se quieren almacenar los datos obtenidos bajo el nombre por defecto de '*Calib_Results_Stereo.mat*'.

```

Stereo calibration parameters after loading the individual calibration files:

Intrinsic parameters of left camera:

Focal Length:      fc_left = [ 86.60096   86.25099 ] ± [ 1.81178   2.08314 ]
Principal point:    cc_left = [ 76.59342   54.03139 ] ± [ 2.49751   2.38213 ]
Skew:              alpha_c_left = [ 0.00000 ] ± [ 0.00000 ] => angle of pixel axes = 9
Distortion:         kc_left = [ -0.07063  -0.05154  -0.00794  -0.00255  0.00000 ]

Intrinsic parameters of right camera:

Focal Length:      fc_right = [ 87.72830   87.54009 ] ± [ 1.94425   2.15068 ]
Principal point:    cc_right = [ 75.93057   55.59944 ] ± [ 3.23399   2.96877 ]
Skew:              alpha_c_right = [ 0.00000 ] ± [ 0.00000 ] => angle of pixel axes =
Distortion:         kc_right = [ -0.09155  -0.00635  -0.00028  -0.00513  0.00000 ]

Extrinsic parameters (position of right camera wrt left camera):

Rotation vector:    om = [ -0.01297   0.97867   0.99121 ]
Translation vector:  T = [ -0.32322  -0.19922   0.21693 ]

```

Figura 126. Resultados de una calibración estéreo

7.3. Obtener datos en 3D de una de las cámaras

Desde esta interfaz también se posibilita la opción de obtener imágenes de profundidad y de amplitud de las cámaras, y mediante los datos de calibración almacenados, hallar la nube de puntos tridimensional obtenida de la escena desde la cámara elegida. Para ello, se despliega el menú de la Figura 127, en el que se puede elegir cuál de las 4 cámaras conectadas al ordenador se quiere utilizar.

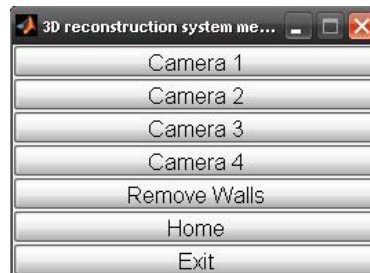


Figura 127. Menú de obtención de datos de las cámaras

Tras la elección de la cámara a utilizar, se pregunta por la carpeta donde se encuentran los parámetros intrínsecos y de distorsión de dicha cámara, y a continuación se muestran en primer lugar la imagen de distancias y de amplitudes (Figura 128), y a continuación la superficie tridimensional capturada (Figura 129, izquierda).

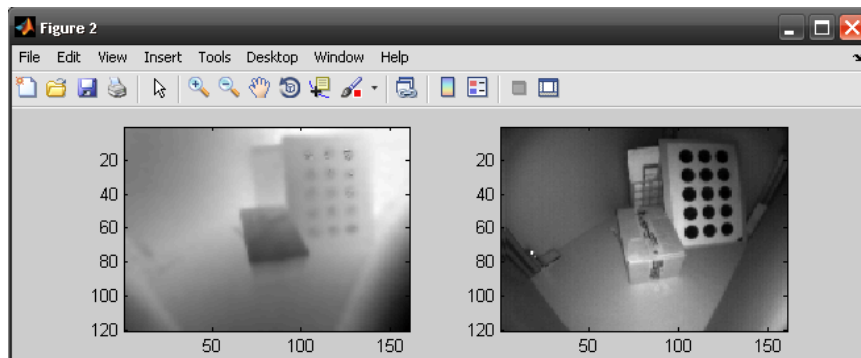


Figura 128. Imagen de distancias (izquierda) y pseudo-imagen de escala de grises (derecha)

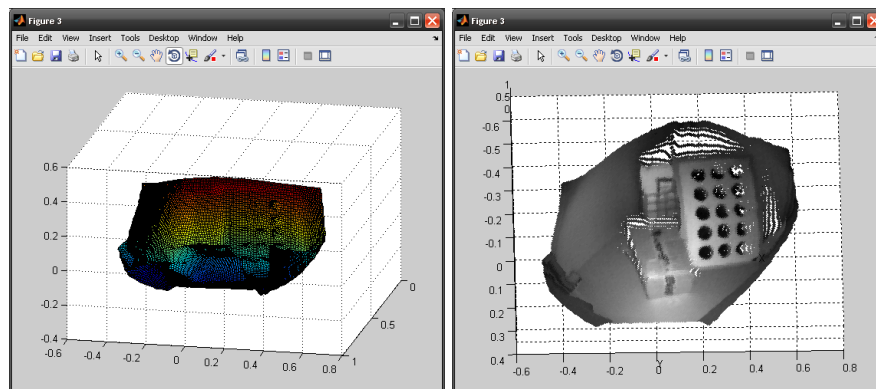


Figura 129. Superficie tridimensional obtenida (izquierda) y con nivel de gris añadido (derecha)

Por último, se pregunta en qué carpeta se quieren almacenar los datos, y se abre por defecto la carpeta '*ObtainedMeshes*'. Por último, se abre un menú que pregunta si se quiere realizar alguna modificación sobre la imagen (Figura 130), como visualizarla junto a su escala de grises (Figura 129 derecha) o eliminar las paredes de la 'caja' en la que se encuentran los objetos de la escena.



Figura 130. Modificaciones sobre los puntos 3D obtenidos por la cámara

7.4. Representar las vistas de 2 de las cámaras

Existe un menú disponible para poder visualizar las nubes de puntos de dos cámaras distintas, haciendo que éstas coincidan mediante su matriz de rotación y traslación estéreo. Se despliega el menú de la Figura 131, en el que se selecciona el par de cámaras que se quiere visualizar. Este apartado está realizado básicamente para poder comprobar la calibración estéreo.



Figura 131. Selección del par de cámaras del que se quiere tomar su nube de puntos

A continuación, se imprime en pantalla en MATLAB® las cámaras seleccionadas, se pide que se seleccionen los archivos en formato .mat que contengan los puntos de cada una de las cámaras y se ha de seleccionar el directorio donde se contienen los datos de calibración estéreo, como se muestra en la Figura 132.

```

Cameras 1 and 2 selected.
LEFT CAMERA: camera 1.
RIGHT CAMERA: camera 2.

Select file containing points from cam1:
Select file containing points from cam2:

Select the path where stereo calibration files are located:

```

Figura 132. Proceso de selección de archivos necesario para reconstrucción 3D estéreo

Una vez seleccionados los ficheros necesarios, se abren dos figuras en las que se representan la nube de puntos de ambas cámaras (Figura 133, izquierda) y las superficies de ambas solapadas (Figura 133, derecha). También se muestran los sistemas de referencia de cada cámara, así como una leyenda en con el color de representación de los datos de cada cámara. De esta manera, se puede comprobar si los datos obtenidos en ambas cámaras coinciden aceptablemente, pudiendo concluir que la calibración estéreo se realizó apropiadamente.

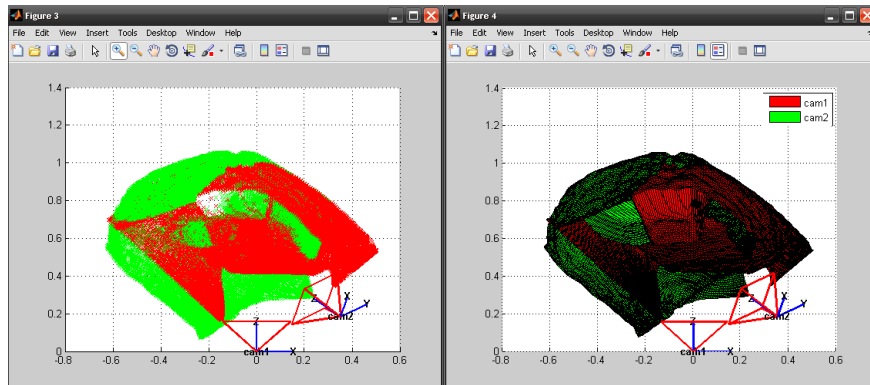


Figura 133. Vista 3D de la escena obtenida por dos cámaras ToF. Izquierda: nube de puntos. Derecha: superficie continua

En caso de que se considere que la calibración ha sido correctamente realizada, pero los datos obtenidos presentan un ligero error que no puede ser corregido realizando otra calibración que sea más precisa, y se quieren hacer coincidir totalmente los puntos 3D, se puede aplicar el algoritmo ICP para hacer que las superficies coincidan en mayor medida. Por lo tanto, tras mostrar las dos reconstrucciones mostradas en la Figura 133, se pregunta al usuario si quiere aplicar ICP. De ser así, se obtienen las 2 nuevas representaciones de la Figura 134.

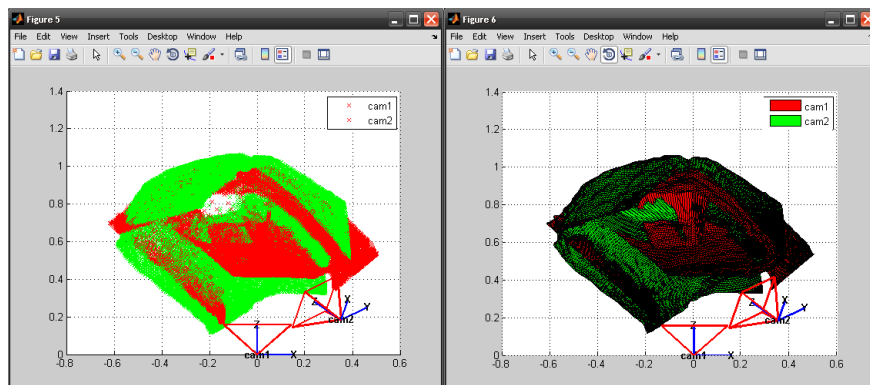


Figura 134. Vista 3D de dos cámaras tras aplicar ICP. Izquierda: nube de puntos. Derecha: superficie continua

7.5. Visualizar la figura con las vistas de las 4 cámaras

Por último, existe una opción para poder representar la escena capturada desde las 4 cámaras disponibles. Para ello, previamente se deben haber capturado las escenas mediante la opción explicada en el apartado 7.3 y almacenado en su correspondiente directorio, y el programa pedirá al usuario que se seleccionen en una ventana emergente y por orden desde la cámara 1 a la 4 los archivos que contienen la escena de cada una de las cámaras, como aparece en la Figura 135:

```
Select file containing points from camera1:
Select file containing points from camera2:
Select file containing points from camera3:
Select file containing points from camera4:
All files correctly load.
```

Figura 135. Cargar en memoria los ficheros con los puntos 3D de cada cámara

A continuación, se va a pedir, de la misma manera (Figura 136) que se carguen los archivos .mat que contienen los datos de la calibración estéreo de cada par de cámaras.

```
Select path containing stereo calibration of cameras 12:
Select path containing stereo calibration of cameras 23:
Select path containing stereo calibration of cameras 34:
Select path containing stereo calibration of cameras 41:
All files correctly load.
```

Figura 136. Cargar en memoria los ficheros de calibración de cada cámara

Por último, de igual forma que ocurría en la visualización estéreo, se abren dos figuras en MATLAB® que contienen la representación de la escena a partir de las 4 cámaras (Figura 137) en forma de nube de puntos y de superficie continua. Además, también se proporciona la opción de aplicar el algoritmo ICP para conseguir que los puntos de las 4 cámaras encajen en mayor medida, como se observa en la Figura 138.

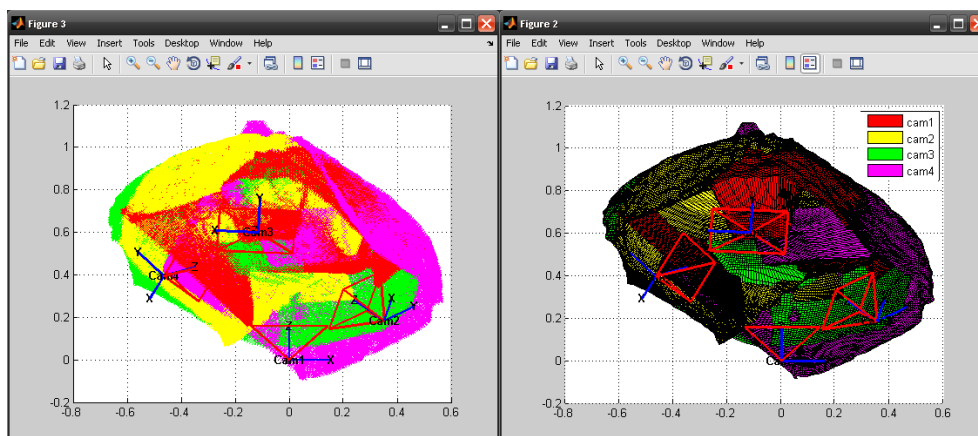


Figura 137. Vista 3D de la escena de las 4 cámaras ToF. Izquierda: nube de puntos. Derecha: superficie continua

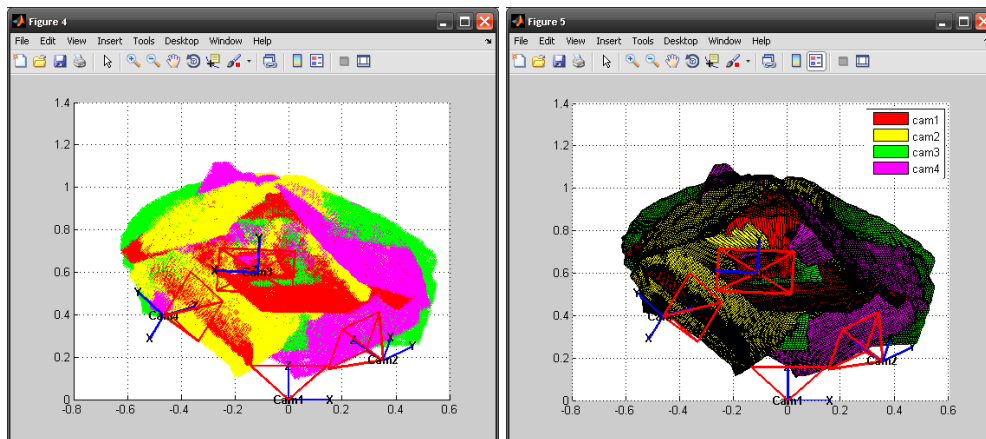


Figura 138. Vista 3D de las 4 cámaras tras aplicar ICP. Izquierda: nube de puntos. Derecha: superficie continua

8. Conclusiones y trabajos futuros

8.1. Conclusiones

A lo largo de este Trabajo Fin de Máster se ha propuesto un sistema de calibración de cámaras de tiempo de vuelo de baja resolución (120 x 160 píxeles) y baja calidad. El primer paso a realizar para ello es la detección de los puntos característicos de un patrón capturado en varias imágenes con distintas posiciones, por lo que debe elegirse un patrón cuyas características sean grandes, para poder ser detectadas aunque el patrón se encuentre alejado del sensor, y además que estén suficientemente distanciadas entre sí para que no se observen como un solo contorno al realizar el tratamiento de la imagen. Para ello, se observa que el patrón damero (*chessboard*), que es el que se utiliza en MATLAB® junto con funciones de detección de esquinas, no proporciona buenos resultados independientemente del tamaño de los cuadros, y tampoco lo hace utilizando las funciones análogas de calibración en OpenCV. Por lo tanto, ha de buscarse otro tipo de patrón válido, así como un algoritmo para buscar puntos de control en él con una precisión sub-pixélica. Se demuestra que un patrón con círculos impresos sobre él cuyas posiciones sean conocidas posibilita la detección precisa de sus centroides por debajo del píxel, por lo que se va a adoptar este patrón con su correspondiente algoritmo de búsqueda de puntos de control diseñados. Por otro lado, a la hora de calibrar y de corregir la distorsión presente en las imágenes es aconsejable tener un alto número de puntos característicos a detectar en cada imagen, pero esto es incompatible con la resolución de la cámara, por lo que se debe llegar a un patrón que cumpla una solución de compromiso entre ambas.

En cuanto a la calibración estéreo, se realiza la correspondencia de puntos entre los pares de imágenes a través del algoritmo diseñado de búsqueda de centroides, y a partir de ahí se calculan las matrices de rotación y traslación de las cámaras en parejas mediante la función de calibración estéreo de la toolbox. Con ello, se analizan los datos obtenidos y se comprueban mediante reconstrucción de puntos 3D, llegando a la conclusión de que finalmente se han obtenido parámetros de calibración válidos a través de las pseudo-imágenes de grises.

Por último, se toman los datos de calibración obtenidos para cada cámara junto con sus medidas de distancia y se calcula la superficie tridimensional que captura cada una de ellas. Se observa que existe un error en la medida de distancia, ya que el valor que devuelven las cámaras no coincide con el real, y esto provoca que las coordenadas 3D de la superficie no coincidan con las reales. Además, este error de medida de distancia no es lineal, sino que varía en función de la propia distancia, por lo que sería necesario realizar otro proceso de calibración añadido para poder obtener una superficie volumétrica que se ajuste de manera precisa a la realidad.

Debido a que cada una de las cámaras presentará su propio error de distancia, a pesar de que se apliquen los parámetros estéreo correctamente calculados para hacer coincidir nubes de puntos de más de una cámara, los puntos no llegarán a coincidir totalmente, ya que se acumula el error de cada una de las cámaras, y esto desemboca en que el resultado de la fusión de datos de varias cámaras no sea satisfactorio, ya que se presentan errores que hay que considerar. Para resolver esto, solamente quedaría la opción de calibrar en distancias. Sin embargo, con el objetivo de conseguir una reconstrucción volumétrica orientativa de la escena, aunque no coincidan las medidas calculadas con las reales, se fijan algunas medidas para tratar de poder reconstruir. Para ello, en primer lugar se calcula un factor de corrección medio de la distancia de

cada una de las cámaras, y además se utiliza una función iterativa ICP (*Iterative Closest Points*) que busca aplicar una nueva rotación y traslación a las nubes de puntos para lograr que coincidan en mayor medida. Con ello se consigue, por lo tanto, una visión orientativa de la escena.

No obstante, las cámaras que se han adquirido son un prototipo de muy bajo coste que el fabricante vende indicando que son útiles para aplicaciones sencillas de usuario en la que la precisión necesaria no sea demasiado alta, por lo que los resultados obtenidos coinciden con los esperados. Sin embargo, en comparación con el resto de cámaras existentes en la actualidad en el mercado es realmente útil embarcarse en este tipo de procesos para lograr la escena 3D con cámaras de bajo coste, ya que el presupuesto del sistema disminuye un orden de magnitud (cámaras de 600€ vs cámaras de más de 3000€).

8.2. Trabajos futuros

Como principales trabajo futuro cabría destacar las siguientes posibilidades:

- La realización de un proceso de calibración de las imágenes de distancias para poder corregir el error que presentan las cámaras, y de esta manera conseguir que la fusión de datos de varias cámaras sea posible y precisa, logrando una reconstrucción volumétrica consistente. Conseguir un nuevo modelo que incluya la profundidad como parámetro, y se calibre a partir de las pseudo imágenes de gris y la imagen de profundidades.
- También podrían realizarse mejoras en la detección de centroides en el patrón, como por ejemplo automatizar aún más la detección de elipses estableciendo los parámetros de búsqueda en función del tamaño y la forma que tenga el patrón en la imagen capturada, lo cual depende de lo lejos o cerca que esté situado, así como la orientación que presente con respecto a las cámaras.
- A pesar de que se puede adjuntar a la representación los tonos obtenidos de la imagen de pseudo-escala de grises, podría ser útil utilizar otra cámara auxiliar móvil RGB que capture la escena desde varios ángulos y así obtener información en color de la escena que se pueda añadir a la reconstrucción 3D.

9. Bibliografía

- [1] Dal Mutto, C., Zanuttigh, P., & Cortelazzo, G. M. (2012). *Time-of-Flight Cameras and Microsoft Kinect™*. Springer.
- [2] Marder, M. (2005). *Comparison of Calibration Algorithms for a Low-Resolution, Wide Angle, 3D Camera*. Master of Science Thesis, Stockholm, Sweden.
- [3] B. Fisher, R (2011). *Sub-pixel estimation*. University of Edinburgh, Scotland, United Kingdom.
- [4] Zhang, Z. (2000). *A flexible new technique for camera calibration*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 22(11), 1330-1334.
- [5] Zhang, Z. (1999). Flexible camera calibration by viewing a plane from unknown orientations. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on* (Vol. 1, pp. 666-673). IEEE.
- [6] Nakano, K., Okutomi, M., & Hasegawa, Y. (2002). *Camera calibration with precise extraction of feature points using projective transformation*. In Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on (Vol. 3, pp. 2532-2538). IEEE.
- [7] Datta, A., Kim, J. & Kanade, T. (2009) *Accurate Camera Calibration using Iterative Refinement of Control Points*, Workshop on Visual Surveillance (VS), 2009 (held in conjunction with ICCV).
- [8] Rapp, H. (2007). *Experimental and theoretical investigation of correlating TOF-camera systems*. Diploma Thesis in Physics. University of Heidelberg, Düsseldorf, Germany.
- [9] Majumder, I. *The Pinhole Camera and Camera Calibration*. University of California Irvine.
- [10] Lázaro, J.L. *Sistemas Avanzados de Metrología de Precisión con Detectores de Array*. Universidad de Alcalá.
- [11] Marr, D., & Hildreth, E. (1980). *Theory of edge detection*. Proceedings of the Royal Society of London. Series B. Biological Sciences, 207(1167), 187-217.
- [12] Bouguet, Y. Camera Calibration Toolbox for Matlab Documentation.
www.vision.caltech.edu/bouguetj/calib_doc/
- [13] Z. Zhang. Microsoft Easy Camera Calibration Tool.
<http://research.microsoft.com/~zhang/calib/>.
- [14] J. Heikillä. Camera calibration toolbox for Matlab (version 3.0).
<http://www.ee.oulu.fi/~jth/calibr/>.
- [15] R. I. Hartley. *An algorithm for self-calibration from several views*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, pages 908–912, 1994.

- [16] Viala, C. R., Salmerón, A. J. S., & Fernández, R. S. (2003). *Técnicas de calibrado de cámaras*. Journal de Mathematiques Pures et Appliques, Leon.
- [17] More, J. (1977). *The Levenberg-Marquardt Algorithm, Implementation, and Theory*. Numerical Analysis, G.A. Watson, ed., Springer-Verlag.
- [18] Möller, T., Kraft, H., Frey, J., Albrecht, M., & Lange, R. (2005). *Robust 3D measurement with PMD sensors*. Range Imaging Day, Zürich, Germany.
- [19] Zhu, Z. (2005). *Lecture 5 – Stereo Vision*. <http://www-cs.engr.ccny.cuny.edu/~zhu/GC-Spring2005/CSc80000-2-VisionCourse.html>
- [20] Weisstein, Eric W. *Euler Angles*. From MathWorld - A Wolfram Web Resource. <http://mathworld.wolfram.com/EulerAngles.html>
- [21] Fajtl, J. *Rodrigues' rotation formula*. Electron Castle. <http://electroncastle.com/wp/?p=39>
- [22] Belongie, S. (1999). *Rodrigues' rotation formula*. From MathWorld—A Wolfram Web Resource, created by Eric W. Weisstein. <http://mathworld.wolfram.com/RodriguesRotationFormula.html>
- [23] Keller, J. M., Gray, M. R., & Givens, J. A. (1985). *A fuzzy k-nearest neighbor algorithm*. Systems, Man and Cybernetics, IEEE Transactions on, (4), 580-585.
- [24] Alexiadis, D. S., Zarpalas, D., & Daras, P. (2013). *Real-time, full 3-D reconstruction of moving foreground objects from multiple consumer depth cameras*. Multimedia, IEEE Transactions on, 15(2), 339-358.

ANEXO I- Detectores de bordes

Con el objetivo de hacer una revisión general de distintos métodos disponibles para encontrar bordes en una imagen, se pueden dividir en dos categorías principales, con sus correspondientes subclases:

- Detectores basados en primera derivada: se buscan valores máximos de la primera derivada de píxeles de la imagen.
 - Detectores tipo gradiente. El gradiente de un píxel de una imagen $\nabla f(u, v)$ se define según la ecuación (69), donde u y v son los píxeles de la imagen:

$$\nabla f(u, v) = \left[\frac{\partial}{\partial u} f(u, v), \frac{\partial}{\partial v} f(u, v) \right] \quad (69)$$

Se considera que existe borde si la magnitud del gradiente supera un determinado umbral U (expresión (70)), y si pertenece al borde se le da un valor 1, y si no, 0, y a partir de esto se construye la imagen de bordes mediante el gradiente, en la que los píxeles que se consideren borde serán de color blanco, y el resto de la imagen, negro.

$$edges(u, v) = \begin{cases} 255 & \text{si } |\nabla f(u, v)| > U \\ 0 & \text{si } |\nabla f(u, v)| < U \end{cases} \quad (70)$$

Algunos ejemplos son los detectores de *Sobel*, *Prewitt* o *Roberts*.

- Detectores tipo orientación. En este caso, en lugar de tener en cuenta el módulo del gradiente, se tiene en cuenta la orientación de éste, lo que se utiliza no solamente para calcular bordes, sino también para calcular los bordes existentes en una determinada dirección (verticales u horizontales), o incluso con un determinado ángulo específico.
- Detector *canny*. Es un tipo de método para encontrar bordes que se basa en la primera derivada, utilizando la derivada de un filtro gaussiano, y que incorpora dos umbrales distintos para la detección de bordes tanto fuertes como débiles. De manera más detallada, los pasos a seguir en este método son:
 - Se suaviza la imagen mediante la aplicación de un filtro gaussiano de varianza conocida σ , según la ecuación (71):

$$G(u, v) = \frac{1}{2\pi\sigma^2} e^{\left[-\frac{u^2+v^2}{2\sigma^2}\right]} \quad (71)$$

- Se obtiene el gradiente (primera derivada) de $G(u, v)$.
- Se adelgaza el ancho de los bordes hasta lograr que solamente tengan 1 píxel de grosor.
- Se fijan dos umbrales distintos U_1 y U_2 de manera que $U_1 < U_2$. Se binarizan todos los píxeles de la imagen obtenida en el paso anterior de manera que los píxeles mayores que U_2 son declarados píxeles de borde “fuerte” y los píxeles situados entre ambos umbrales se denominan píxeles de borde “débil”.

- Los píxeles de borde fuerte son incluidos directamente en la imagen de bordes finales. Sin embargo, los píxeles de bordes débiles solamente son incluidos si tienen conectividad 8 con algún otro borde incluido.
- Detectores basados en segunda derivada: los píxeles donde hay cruces por cero en la segunda derivada son considerados candidatos de borde.
 - Detector *Laplaciana*. Se halla la segunda derivada $\nabla^2 f(u, v)$, como se muestra en la expresión (72), y se buscan los cortes por cero. Dichos puntos serán los que pertenezcan a los bordes.

$$\nabla^2 f(u, v) = \frac{\partial^2}{\partial u^2} f(u, v) + \frac{\partial^2}{\partial v^2} f(u, v) \quad (72)$$

- Detector *Laplaciana de la Gaussiana*. En primer lugar, se aplica un filtro gaussiano a la imagen (como el de la ecuación (71)) y a continuación se aplica la Laplaciana (expresión (73)), la segunda derivada aplicada al resultado de filtrar la imagen, que será la expresión sobre la que se busquen los cortes por cero.

$$\nabla^2 G(u, v) = -\frac{1}{\pi\sigma^4} \left(1 - \frac{u^2 + v^2}{2\sigma^2} \right) e^{\left[-\frac{u^2+v^2}{2\sigma^2}\right]} \quad (73)$$

ANEXO II: Método Zhang

II.1. Homografía entre el plano modelo y el plano imagen

Partiendo de la base de que se está utilizando un patrón de calibración plano bidimensional, se puede decir que el plano imagen se encuentra situado en $Z = 0$ del sistema de coordenadas del mundo, y por tanto, la ecuación (20) queda de la siguiente manera:

$$\begin{aligned} s \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} &= \begin{pmatrix} f_x & \gamma & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{pmatrix} \begin{pmatrix} X \\ Y \\ 0 \\ 1 \end{pmatrix} = \\ &= \begin{pmatrix} f_x & \gamma & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ r_{31} & r_{32} & t_z \end{pmatrix} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} \end{aligned} \quad (74)$$

O lo que es lo mismo:

$$s\tilde{p} = K \cdot [r_1 \quad r_2 \quad t] \cdot \tilde{P} \quad (75)$$

Convirtiendo esta expresión en una homografía H que relaciona los puntos del espacio tridimensional con sus correspondientes puntos en la imagen, se puede expresar de la siguiente forma:

$$s\tilde{p} = K \cdot [r_1 \quad r_2 \quad t] \cdot \tilde{P} = H \cdot \tilde{P} \quad \text{con} \quad H = K \cdot [r_1 \quad r_2 \quad t] \quad (76)$$

II.2. Restricciones de los parámetros intrínsecos

Dada una imagen del plano modelo, la homografía de la ecuación (76) puede ser estimada. Tomando H por sus columnas, $H = [h_1 \quad h_2 \quad h_3]$, se obtiene:

$$[h_1 \quad h_2 \quad h_3] = \lambda \cdot K \cdot [r_1 \quad r_2 \quad t] \quad (77)$$

Debido a que la matriz R es una matriz de rotación, los vectores que la componen cumplen las restricciones de ortonormalidad [15]:

$$r_1^T \cdot r_1 = r_2^T \cdot r_2 \quad (78)$$

$$r_1^T \cdot r_2 = 0 \quad (79)$$

De las expresiones (82) y (83) se obtiene:

$$h_1^T K^{-T} K^{-1} h_1 = h_2^T K^{-T} K^{-1} h_2 \quad (80)$$

$$h_1^T K^{-T} K^{-1} h_2 = 0 \quad (81)$$

Por lo tanto, dada una homografía, se fijan dos restricciones de los parámetros intrínsecos (ecuaciones (80) y (81)). Como una homografía tiene 8 grados de libertad y hay 8 parámetros extrínsecos (3 de rotación y 3 de traslación), solamente se pueden obtener dos restricciones de los intrínsecos.

Para resolver esta homografía, se utilizará la ecuación (76) como se muestra a continuación:

$$s\tilde{p} = H \cdot \tilde{P} \quad \rightarrow \quad s \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \cdot \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} \quad (82)$$

Desarrollando la ecuación (82), se obtienen las siguientes expresiones:

$$u = h_{11} \cdot X + h_{12} \cdot Y + h_{13} \quad (83)$$

$$v = h_{21} \cdot X + h_{22} \cdot Y + h_{23} \quad (84)$$

$$1 = h_{31} \cdot X + h_{32} \cdot Y + h_{33} \quad (85)$$

Dividiendo las ecuaciones (83) y (84) entre la ecuación (85), se obtienen:

$$u = \frac{h_{11} \cdot X + h_{12} \cdot Y + h_{13}}{h_{31} \cdot X + h_{32} \cdot Y + h_{33}} \quad (86)$$

$$v = \frac{h_{21} \cdot X + h_{22} \cdot Y + h_{23}}{h_{31} \cdot X + h_{32} \cdot Y + h_{33}} \quad (87)$$

Que también se puede expresar según las siguientes ecuaciones:

$$h_{11} \cdot X + h_{12} \cdot Y + h_{13} - h_{31} \cdot u \cdot X - h_{32} \cdot u \cdot Y - h_{33} \cdot u = 0 \quad (88)$$

$$h_{21} \cdot X + h_{22} \cdot Y + h_{23} - h_{31} \cdot v \cdot X - h_{32} \cdot v \cdot Y - h_{33} \cdot v = 0 \quad (89)$$

Expresando esto de manera matricial de forma $A \cdot b = 0$, con el objetivo de resolverlo mediante descomposición en valores singulares, se obtiene la expresión (90):

$$\begin{pmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_i & Y_i & 1 & 0 & 0 & 0 & -u_i \cdot X_i & -u \cdot Y_i & -u \\ 0 & 0 & 0 & Y_i & Y_i & 1 & -v_i \cdot X_i & -v \cdot Y_i & -v \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix} \cdot \begin{pmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{pmatrix} = \begin{pmatrix} \vdots \\ 0 \\ 0 \\ \vdots \end{pmatrix} \quad (90)$$

Las incógnitas a resolver son cada uno de los parámetros de la matriz H , y por tanto, como hay 9 incógnitas h_{ij} , son necesarias al menos 9 ecuaciones para poder resolver el sistema. Sabiendo que cada punto proporciona 2 ecuaciones (coordenada u y v), serán necesarios al menos 5 puntos por imagen ($2 \cdot 5 = 10$). Es decir, que el patrón ha de tener, como poco, 5 puntos de control, 5 circunferencias para detectar sus centroides. Sin embargo, lo ideal es sobredimensionar el sistema para poder optimizar y tratar de minimizar errores. En este caso, la solución se obtiene a partir de la descomposición en valores singulares, siendo el vector propio asociado al valor propio más pequeño la solución de menor error.

II.3. Resolución de la calibración de la cámara

Para resolver el sistema, primero se da una solución analítica y a continuación se realiza una técnica de optimización no lineal basada en el criterio de máxima verosimilitud. Por último, se calcula la distorsión de la lente.

Definiendo una matriz B como $B = K^{-T}K^{-1}$, se obtiene la ecuación (91):

$$K^{-T}K^{-1} = B = \begin{pmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{pmatrix} =$$

$$= \begin{pmatrix} 1/f_x^2 & -\gamma/f_x^2 f_y & v_o \gamma - u_o f / f_x^2 f_y \\ -\gamma/f_x^2 f_y & \gamma/f_x^2 f_y + 1/f_y^2 & -\gamma(v_o \gamma - u_o f) / f_x^2 f_y - u_o / f_y^2 \\ v_o \gamma - u_o f / f_x^2 f_y & -\gamma(v_o \gamma - u_o f) / f_x^2 f_y - u_o / f_y^2 & (v_o \gamma - u_o f)^2 / f_x^2 f_y + v_o / f_y^2 + 1 \end{pmatrix} \quad (91)$$

Como B es una matriz simétrica, se puede definir con un vector de 6 elementos:

$$b = [B_{11} \ B_{12} \ B_{22} \ B_{13} \ B_{23} \ B_{33}]^T \quad (92)$$

Por tanto, si la columna i -ésima de la matriz H es $h_i = (h_{i1} \ h_{i2} \ h_{i3})^T$, entonces se tiene que:

$$h_i^T B h_i = v_{ij}^T b \quad (93)$$

Donde $v_{ij}^T b$ es:

$$v_{ij}^T = (h_{i1}h_{j1} \ h_{i1}h_{j2} + h_{i2}h_{j1} \ h_{i2}h_{j2} \ h_{i3}h_{j1} + h_{i1}h_{j3} \ h_{i3}h_{j2} + h_{i2}h_{j3} \ h_{i3}h_{j3}) \quad (94)$$

Teniendo esto en cuenta, las dos restricciones fundamentales de una homografía de las ecuaciones (80) y (81) pueden ser reescritas según la ecuación (95):

$$\begin{bmatrix} v_{11}^T - v_{22}^T \\ V_{12}^T \end{bmatrix} b = 0 \quad (95)$$

Si se toman n imágenes del patrón o plano modelo, tomando las ecuaciones (95) de cada una de ellas, se tiene:

$$V \cdot b = 0 \quad (96)$$

Donde V es una matriz de $2n \times 6$. Si n es mayor o igual que 3, podemos tener una única solución de b , que es donde están contenidos los parámetros que queremos hallar. Si n es igual a 2, se puede hallar también, pero fijando que el skew va a ser nulo ($\gamma = 0$). Por tanto, el mínimo número de imágenes es 2 ó 3, según se considere despreciable el skew o no. Sin embargo, normalmente se suele sobredimensionar el sistema.

Una vez que b es calculado mediante descomposición en valores singulares, se pueden hallar los parámetros intrínsecos de la cámara según las siguientes ecuaciones:

$$v_o = \frac{(B_{12}B_{13} - B_{11}B_{23})}{(B_{11}B_{22} - B_{12}^2)} \quad (97)$$

$$\sigma = B_{33} - \frac{(B_{13}^2 + V_o(B_{12}B_{13} - B_{11}B_{23}))}{B_{11}} \quad (98)$$

$$f_x = \sqrt{\frac{\sigma}{B_{11}}} \quad (99)$$

$$f_y = \sqrt{\frac{\sigma B_{11}}{(B_{11}B_{22} - B_{12}^2)}} \quad (100)$$

$$\gamma = -\frac{B_{12}f_x^2 f_y}{\sigma} \quad (101)$$

$$u_o = \frac{\sigma y_o}{f_x} - \frac{B_{13}f_x^2}{\sigma} \quad (102)$$

Una vez conocidos los parámetros intrínsecos de la matriz K , se pueden calcular los parámetros extrínsecos para cada imagen según las ecuaciones que aparecen a continuación:

$$r_1 = \frac{1}{\|K^{-1}h_1\|} K^{-1}h_1 \quad (103)$$

$$r_2 = \frac{1}{\|K^{-1}h_2\|} K^{-1}h_2 \quad (104)$$

$$r_3 = r_1 \times r_2 \quad (105)$$

$$t = \left(\frac{1}{\|K^{-1}h_1\|} + \frac{1}{\|K^{-1}h_2\|} \right) \cdot \frac{K^{-1}h_3}{2} \quad (106)$$

II.4. Estimación de máxima verosimilitud

Con el objetivo de disminuir el error que pueda existir del proceso anterior y de refinar los resultados, se va a aplicar un proceso iterativo basado de la técnica de máxima verosimilitud. Dadas n imágenes del patrón en la escena con m puntos de control, y asumiendo que el ruido que afecta a las imágenes es independiente e idénticamente distribuido, la estimación de máxima verosimilitud puede ser obtenida minimizando la ecuación (107):

$$\sum_{i=1}^n \sum_{j=1}^m \|p_{ij} - \hat{p}(K, k_1, k_2, R_i, t_i, P_j)\|^2 \quad (107)$$

Donde $\hat{p}(K, k_1, k_2, R_i, t_i, P_j)$ es la proyección de un punto P_j según la ecuación (76) y, con los parámetros de distorsión radiales k_1 y k_2 añadidos, según fueron explicados en la sección 2.3. Modelo Pinhole y parámetros de distorsión. La ecuación (107) se puede resolver mediante el método Levenberg-Marquardt. Este método necesita una estimación inicial de los parámetros K , R_i y T_i , y para ello se va a utilizar lo obtenido en las ecuaciones (97) - (106). Una vez finalizado el algoritmo iterativo, tendremos finalmente los parámetros intrínsecos y extrínsecos de la cámara refinados y como resultado final de calibración.

Éste es el proceso que realiza el método Zhang. Sin embargo, en la toolbox de MATLAB® de Yves-Bouguet basada en el método Zhang se ha añadido también el cálculo de los parámetros de distorsión tangencial p_i , por lo que la función a minimizar es la (108):

$$\sum_{i=1}^n \sum_{j=1}^m \|p_{ij} - \hat{p}(K, k_1, k_2, p_1, p_2, p_3, R_i, t_i, P_j)\|^2 \quad (108)$$

ANEXO III. Geometría epipolar

Una manera sencilla de explicar un sistema de visión estéreo es mediante la geometría epipolar. La geometría epipolar [19] es la geometría proyectiva intrínseca entre dos vistas distintas y es se puede utilizar con el fin de obtener la posición 3D de un punto a partir de la búsqueda de sus correspondientes parejas de puntos en cada una de las imágenes estéreo.

En la Figura 139 se muestran las bases de la geometría epipolar, en la que un punto P tridimensional es representado por los vectores $P_L = (X_L, Y_L, Z_L)$ y $P_R = (X_R, Y_R, Z_R)$ referido a cada uno de los sistemas de referencia de cada cámara respectivamente (izquierda y derecha). La proyección de P en el plano imagen de la cámara izquierda sería $p_l = (u_l, v_l, z_l)$ y en la derecha $p_r = (u_r, v_r, z_r)$, sabiendo que la coordenada z de los puntos proyectados coincide con la distancia focal de cada una de las cámaras.

Los epipolos son las imágenes proyectadas en una cámara del correspondiente centro óptico de proyección de la otra cámara (e_l y e_r). El plano formado por el centro de proyección de cada una de las cámaras (O_l y O_r) y el punto tridimensional P se llama plano epipolar. Las rectas epipolares son las formadas por la intersección del plano epipolar con cada plano imagen. Además, estas rectas también son las que unen el epipolo (e_l o e_r) y con su correspondiente proyección 2D del punto P (p_l o p_r).

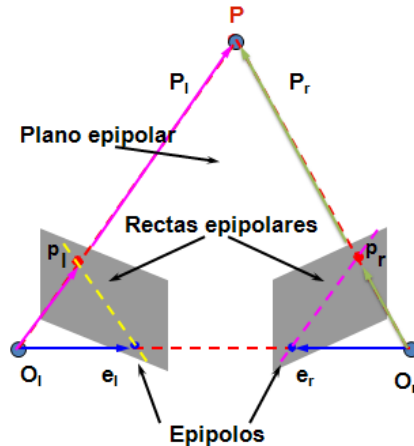


Figura 139. Geometría epipolar¹³

Esta geometría es muy útil para ayudar a solventar el problema de búsqueda de correspondencia de puntos en ambas imágenes debido a que siempre ha de cumplirse esta restricción: dado un punto p_l de la cámara izquierda, su homólogo en la cámara derecha va a estar situado en su recta epipolar conjugada del plano imagen derecho, y viceversa. Existe una matriz llamada ‘matriz fundamental’ o matriz F y que cumple la siguiente condición:

$$p_R^T \cdot F \cdot p_L = 0 \quad (109)$$

Por lo tanto, conociendo la matriz F podríamos hallar puntos correspondientes entre ambas imágenes. La matriz F se puede calcular según la expresión (110):

¹³ http://en.wikipedia.org/wiki/File:Epipolar_geometry.svg

$$F = K_R^{-T} \cdot R \cdot S \cdot K_L^{-1} \quad (110)$$

Siendo S la siguiente matriz formada por los parámetros del vector de traslación T , siendo $T = (T_X, T_Y, T_Z)^T$ el vector que mide la distancia entre los centros de proyección de ambas cámaras:

$$S = \begin{pmatrix} 0 & -T_Z & T_Y \\ T_Z & 0 & -T_X \\ -T_Y & T_X & 0 \end{pmatrix} \quad (111)$$

Por lo tanto, si se calibra la cámara, calculando la matriz F podríamos hallar correspondencia entre puntos de varias imágenes. Sin embargo, aunque es útil y necesario conocer la geometría epipolar, para este trabajo no va a ser necesario realizar esta tarea porque la correspondencia va a venir previamente calculada por el cálculo de los puntos característicos del patrón en las imágenes tomadas por ambas cámaras, como se detalla en la sección de calibración estéreo.

Sin embargo, conociendo las correspondencias de puntos entre ambas imágenes, así como las matrices R y T de la calibración estéreo, podría servir para corroborar que la calibración se ha desarrollado correctamente.